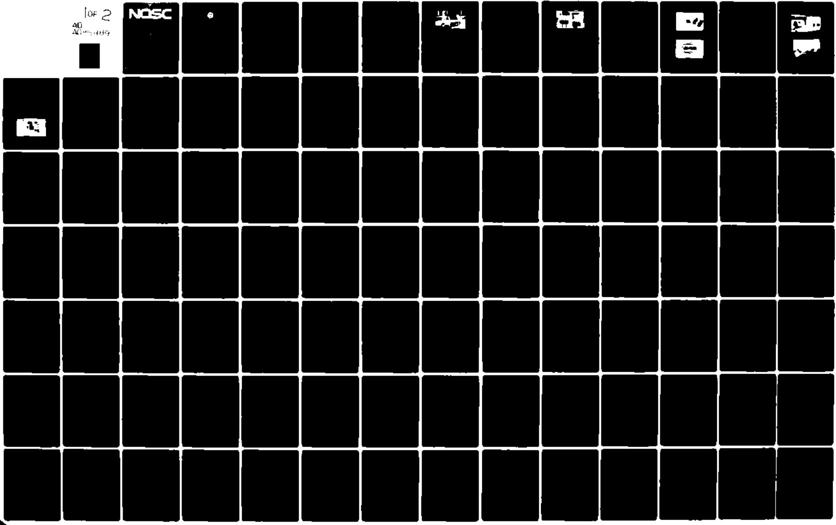
AD-A085 989 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 7/2

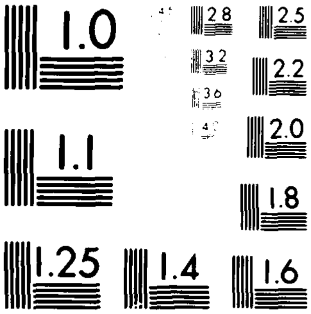
MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM.(U)

SEP 79 C CLAVELL

UNCLASSIFIED NOSC/TR-532 NL

1 OF 2
AD
A085989





MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

NOSC

Technical Report 532

# MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM

C Clavell, Jr

September 1979

Final Report

Prepared for
Naval Material Command

Approved for public release;
distribution unlimited.

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NOSC

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

## ADMINISTRATIVE INFORMATION

The work reported herein was sponsored by the Naval Material Command (63765N, F57572, ZF57572004, 513-ME02) and conducted over the period October 1976 to September 1979.

Released by
S Yamamoto, Head
Marine Sciences Division

Under authority of
HO Porter, Head
Biosciences Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NOSC Technical Report 532 (TR 532)

2. GOVT ACCESSION NO.
AD-A085989

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM

5. TYPE OF REPORT & PERIOD COVERED
9 Final Report, 16
Oct[redacted]76-Sep[redacted]79

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 C. Clavell, Jr
16 F57572

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Ocean Systems Center
San Diego, California 92152

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
17
63765N, F57572, ZF57572004
513-ME02

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Material Command
Technology Div., (NMAT-08T245)
Washington, D.C. 20360

12. REPORT DATE
11 Sept[redacted]79

13. NUMBER OF PAGES
182

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
12 176

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

14 NOSC/TR-532

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A microcomputer has been adapted for use with a flow-through anodic stripping voltammetry (ASV) instrument developed by the Naval Ocean Systems Center. The ASV instrument was originally developed to provide near-real-time analysis of trace metals in seawater. Four years' experience with this system in the coastal waters of Florida, Peru, and California showed that many ancillary instruments, such as pH meters, specific ion meters, and fluorometers, could be added to better define sharply varying ocean environments. To coordinate the operations of these instruments and to deal with the very large amount of raw data that would be generated,

(Continued)

393159

20. Abstract (Continued)

a Motorola 6800 microcomputer was added. Two major system software routines were developed for this application. The first controls ASV instrumentation, while the second is a monitor program that enables the system to function as a "stand alone" microcomputer. Thus equipped, the system should greatly facilitate trace metal investigations in the field by increasing the amount and accuracy of the data obtained while decreasing routine manual operations. Moreover, with substitution of a simulator for the wet-chemistry electromechanical unit, the system can serve as a tool for developing new software and hardware without requiring an investment in new, unproven equipment. The system can be expanded or modified within constraints of size, weight, and available memory.

## SUMMARY

A second-generation, microcomputer-controlled flow-through anodic stripping voltammetry (ASV) instrument has been developed by the Naval Ocean Systems Center (NOSC). The first-generation, non-computerized instrument was originally developed to provide near-real-time analysis of trace metals in seawater. Four years' experience with this system in the coastal waters of Florida, Peru, and California indicated the need for added instrumental flexibility to permit changes in methodologies and to deal with the very large amount of raw data generated. To achieve these goals, a new system designed around a Motorola 6800 microcomputer was developed. Two major system software routines were written for this application: an instrument control program and a computing routine. The latter performs the necessary calculations to yield actual concentration values for each metal studied. Thus equipped, the system has improved the amount and accuracy of the data obtained while decreasing routine manual operations. Moreover, with the substitution of a simulator for the wet-chemistry electromechanical unit, the system can serve as a tool for developing new software and hardware without requiring investment in new, unproven equipment. The system can be expanded or modified within the constraints of size, weight, and available memory.

## BACKGROUND

ASV is a form of polarography that has produced new interest in this field in recent years (Refs. 1-3). It is a technique that allows relatively fast measurements, to be made for various metals in different solutions without the need for elaborate preparations or preconcentration procedures. The technique's sensitivity is outstanding, having approached the parts-per-trillion level for several metals.

From the standpoint of field measurements, the main metals of interest are copper, lead, cadmium, and zinc. Using present equipment and techniques, it is possible to make simultaneous measurements of any three of these four metals directly in seawater with no pretreatment. It is this feature which has made ASV such an attractive tool for research and routine monitoring operations.

A generalized ASV instrument consists of (1) a potentiostat that controls the potential on the working electrode during electrolysis; (2) a cell arrangement to hold the sample and electrodes; and (3) some type of data display device, generally an X-Y recorder. Figure 1 illustrates a typical commercial system. Although there are several types of materials commonly used for the working electrode, this discussion will be limited to those composed of solid, glassy carbon. This electrode is most frequently rod-shaped, and either a short section of the rod or simply the face of one end is used as the active electrode surface. For either case, the surface to be use must be highly polished.

A generalized procedure for an ASV measurement of Cu, Pb and Cd begins with the application of a negative potential (vs SCE reference electrode) to the working electrode. The working electrode is in contact with the sample solution, to which a dilute mercury solution has been added. The applied potential causes a thin mercury metal film to be plated onto the electrode surface, and subsequently the metal ions in the sample are reduced and plated into the mercury film for several minutes. It is this electrolytic concentration which gives the technique its great sensitivity. After an appropriate time interval, the electrode is "scanned" by incrementing the potential in the positive direction. As the

LRO 4812-8-77B

Figure 1. Commercially available system for ASV.

electrode voltage becomes less negative, the individual metals oxidize out of the mercury film, causing a current to flow. The current can be detected and displayed as a peak on the strip chart recorder (Fig. 2). Since each metal oxidizes out of the film at its own characteristic potential, it is possible to effect excellent separations in the multicomponent samples, with quantification being made by the technique of standard additions (Ref. 4).

There are many variations in both the methodology and the equipment currently used for ASV. Recently, new equipment incorporating microprocessors to control instrument parameters and aid in the display of results has come on the market. The Princeton Applied Research Model 374-1 is one example. Other instruments utilizing hard-wired programming designed for system flexibility have also been introduced recently, an example being Environmental Science Association's Model 3040. However, there is one trait common to all systems that are presently available commercially. They are basically laboratory instruments geared to measuring individual samples.

For the researcher interested in real-time events (biological phenomena, pollution surveys, etc.), such equipment is of little value, since the need to collect and process individual samples inherently precludes real-time analysis. An additional problem, which becomes extremely significant at the parts-per-billion (ppb) level, is sample contamination resulting from human intervention in the sample acquisition and during secondary handling at the man-instrument interface.

To overcome these problems, a flow-through ASV instrument system was developed at NOSC that provides near-real-time capability and requires no operator intervention in sampling (Ref. 6). As Fig. 3 shows, the system components are housed in four containers designed for field use. The potentiostat, the hard-wired programming unit, and the strip chart recorder are packaged in individual boxes, while the fourth box contains the chemical, electromechanical cell. The system can run unattended for periods of up to 48 hours, but suffers from lack of hardware flexibility. Thus, modification or expansion of the system is

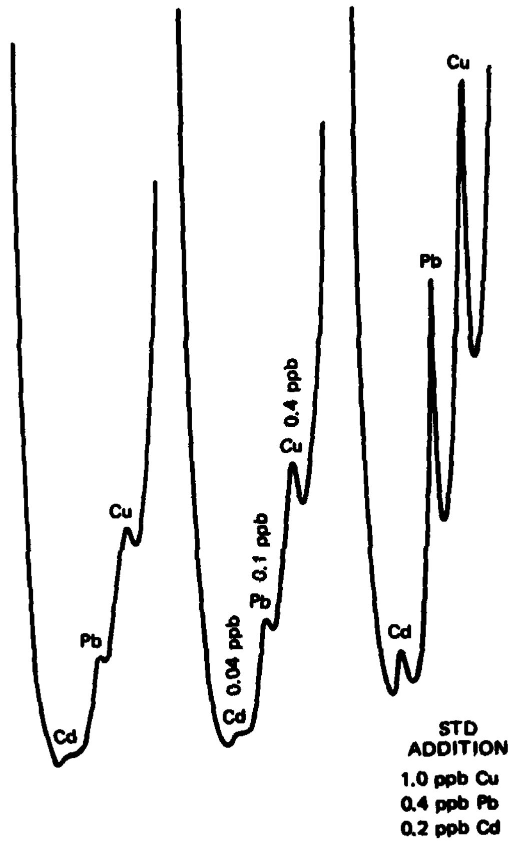


Figure 2. Typical traces from ASV.

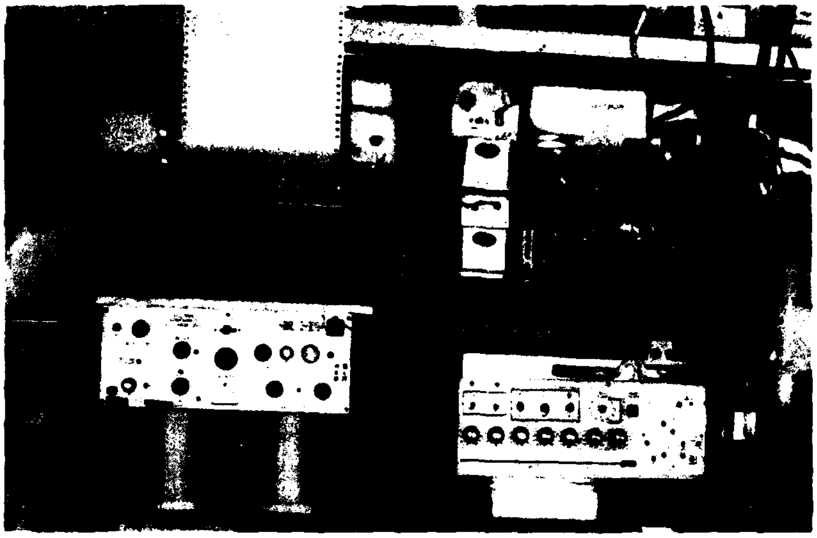


Figure 3. Automated real-time chemical analysis equipment.

very difficult. This particular instrument has been in service for approximately 4 years and was used in the coastal waters of Florida, Peru, and California.

The type of information provided by the flow-through instrument may be seen in Fig. 4. This figure shows the results of a 5-day, continuous operation conducted to measure zinc levels at a stationary point in San Diego Bay. The lower curves, consisting of 358 individual measurements, represent the raw data of relative peak current vs time; the upper trace is the processed data, giving the actual Zn concentrations in ppb vs time. It can be seen from these data that tidal fluctuations have a profound influence on the zinc concentration at any given moment and consequently could lead an investigator to draw erroneous conclusions about the zinc content of the bay if sampling had occurred at more widely spaced intervals.

In principle, this instrument operates like the general ASV instrument described earlier. Functionally, however, the instrument is unique and represents a significant advancement in the state of the art of trace-metal investigations in the field.

The major innovations consisted of the development of the wax-impregnated tubular graphite electrode (Ref. 5) (which has recently been supplanted by the flow-through carbon disk electrode) and the multielement tubular reference electrode (Ref. 6). Both have been patented by NOSC. The wax-impregnated tubular graphite electrode consists simply of a 1/4-in. high-purity graphite rod with a 1/8-in. hole drilled through the center. Before use, the graphite tube is vacuum-impregnated with paraffin, and the inner bore is sanded and polished to a high gloss (Fig. 5). A newly developed sensor, the carbon disk electrode (Fig. 5), consists of a Teflon holder that can accommodate two glassy carbon or LTI (low-temperature isotropic) carbon disks. Each disk is approximately 5/8 in. in diameter and 1/16 in. thick. As with the tubular graphite electrode, electrical contact with the disks is made by a Pt wire lead pressed against the surface. The multielement reference electrode (Fig. 6) incorporates an Ag/AgCl element and a platinum element within the same tubular

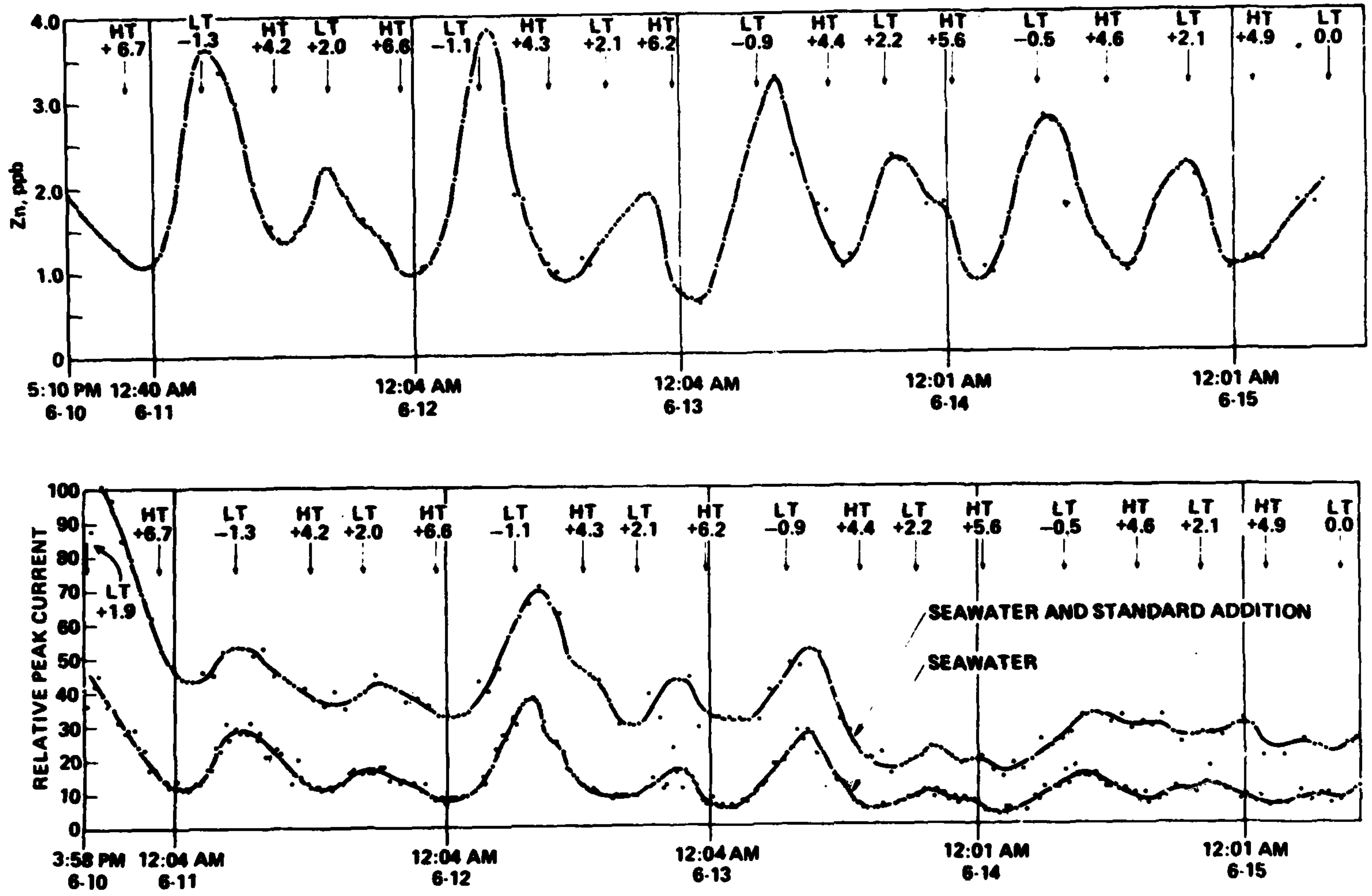
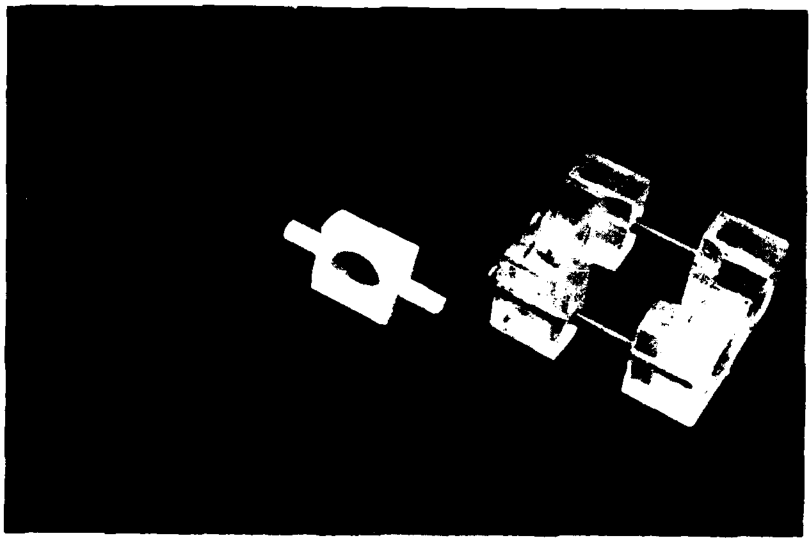


Figure 4. Zinc tracings made using flow-through instrument. Upper figure: Tidal fluctuations of Zn concentrations in San Diego Bay during 10–15 June 1975. Lower figure: Peak currents for Zn in seawater and seawater plus standard addition.



LRO 4816-8-77B

Figure 5. Working electrodes for flow-through instrument.

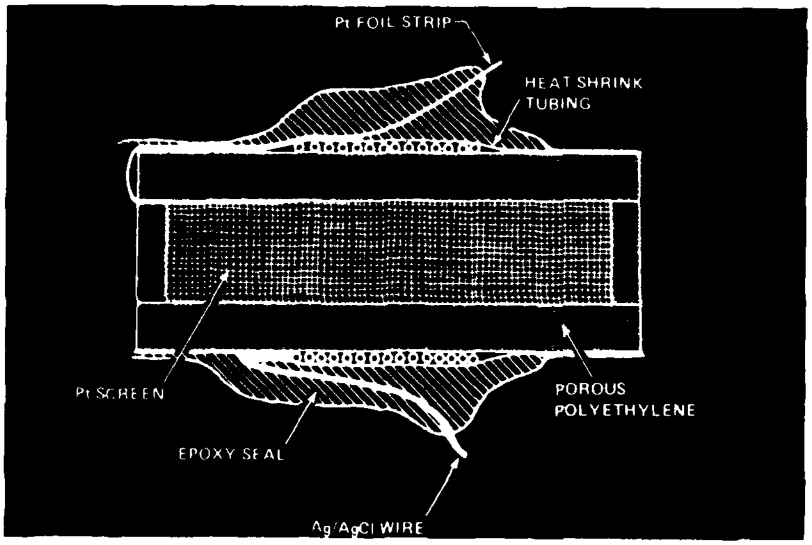


Figure 6. Reference electrode for flow-through instrument.

electrode. Isolating the Ag/AgCl element from the sample stream with a salt bridge has eliminated the problem of $Ag^+$ ion contamination.

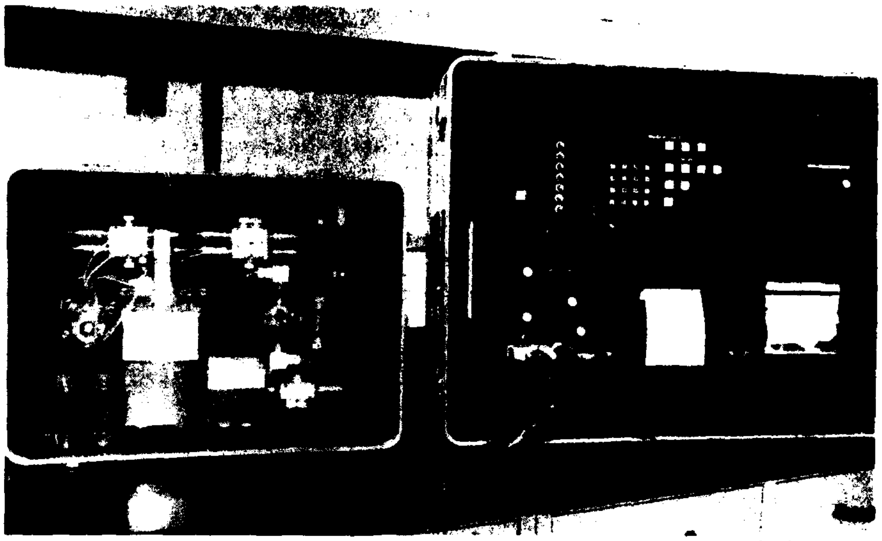
The initial success of the flow-through instrument prompted the design and construction of a new version utilizing a microcomputer. The decision to use a microcomputer was based on experience with the previous instrument, which clearly showed the great need for flexibility in such a system in order to cope with various environments in which the instrument was used as well as to take advantage of new analytic methodologies that might become available. Also, as the system evolved and more equipment was added, such as pH meters, specific ion meters, and fluorometers, the amount of raw data generated became unwieldy, requiring months of processing following a sea cruise.

The use of a microcomputer as both system controller and data manipulator facilitated data handling and increased flexibility. The microcomputer permits a great deal of latitude in system hardware configuration. The instrument is now able to incorporate ancillary instruments and coordinate the different functions of the total system. Because programs can be easily changed, modified, or updated, the instrument hardware package can evolve fresh capabilities as new equipment, sensors, or techniques are developed. An example of such a modification would be the addition of an auxiliary reservoir and associated valves to make it possible to alter the medium present in the electrodes during the scanning cycle. This procedure for stripping or scanning into a different electrolyte permits analysis of metals other than Cu, Pb, Cd, and Zn. For example, by stripping into a $HC10_4$ solution, the instrument can be used for the detection of mercury in seawater (Ref. 7).

A further benefit derived from use of a microcomputer is the powerful data processing capability of the instrument. Thus, it is now possible to achieve real-time data reduction, permitting parameter adjustments to the system based on the processed data displayed.

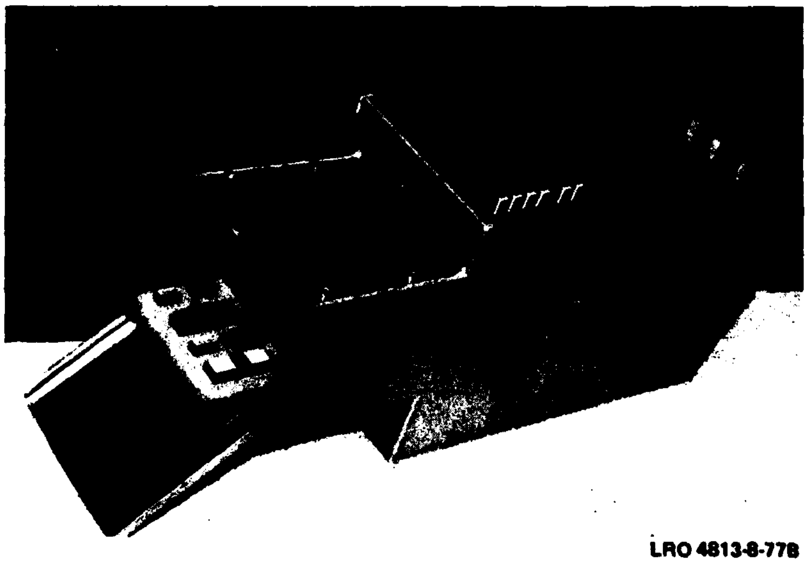
## HARDWARE

To facilitate alterations to the system, the microcomputer section in the instrument has been made readily accessible to the user. Thus, there are two major functional modes of operation: the dedicated ASV instrument system, with its programs on EPROMS (erasable programmable read-only memories), and an accessible, independent microcomputer. The current electronic configuration (Figs. 7 and 8) utilizes a Motorola 6800 microcomputer with 16K words of RAM (random-access memory) and sockets for 32K words of EPROMS. The control functions are implemented through eight peripheral interface adaptors with a total of 128 I/O lines, which can be programmed as inputs or outputs, and 32 interrupt lines, half of which can be programmed as inputs or outputs. Interfacing to high-current components such as motors and solenoids is effected through solid state relays and transistor drivers. Two independent clocks are available. One is used for time of day as well as some timing sequences, while the other is a digital-panel-meter elapsed timer that can be programmed to display any desired number of seconds and countdown, giving an output pulse as the zero time indication. The elapsed time provides the operator with a visual countdown, allowing him to monitor the time remaining during any cycle of a sequence as well as signalling the computer when the timing is complete. Data input to the computer is via a 16-key keyboard. Experimental parameters, such as cycle times and voltage values, are input in decimal form and converted to BCD (binary coded decimal) and binary formats for machine use. Computed results are presented on a 40-column alphanumeric printer, while the raw data is printed on a 5- or 10-in. strip chart recorder. See Appendix B for complete system schematics.



LRO 4815-8-77B

Figure 7. Microcomputer-assisted flow-through ASV system.



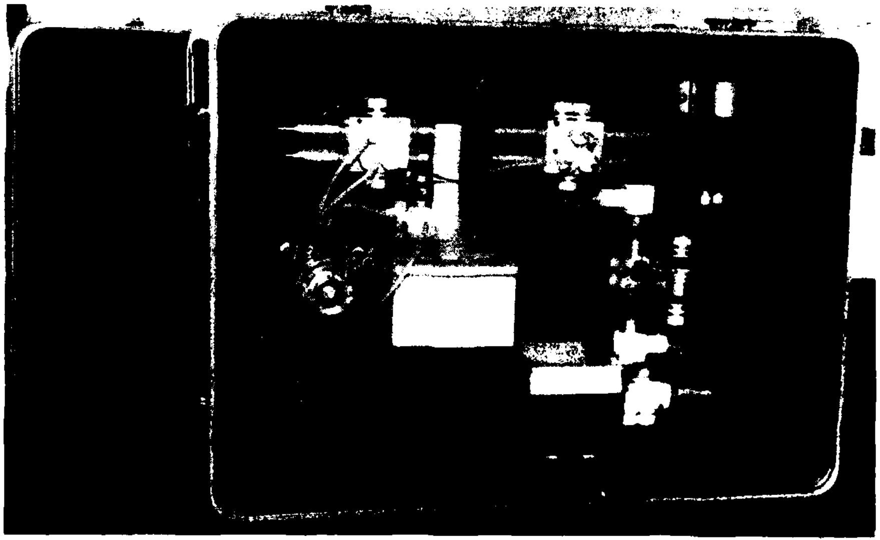
LRO 4813-8-77B

Figure 8. Microcomputer component of new flow-through ASV system.

The chemical cell, as shown in Fig. 9, consists of five custom Teflon solenoid valves, sample and mercury reservoirs, pumps, and the necessary plumbing. Fig. 10 is a diagrammatic representation of the chemical package. It illustrates the interconnections between the valves, pumps and reservoirs. Fig. 10 should be used for reference in the following example of a single analysis sequence. The sequence begins by switching valves 1 and 2 to the A site and activating pump number 1, causing the mercury solution to be circulated through the electrodes.

A negative potential (vs the reference) is now applied to the carbon working electrode for approximately 6 s, causing a thin film of mercury to be plated on the carbon electrode's surface. At the end of this time, valves 1 and 2 are switched to the B side and the sample is now pumped through the electrodes. The sample may either be recirculated or discarded, depending on the setting of valve 3. In this example, the sample is discarded after making one pass through the electrodes. The potential applied previously to the mercury solution is maintained, and the 1-liter sample is plated for 4 min. When the sampling is completed, pump 1 deactivates, and the potential scan of the electrode begins. The peak current data collected at this time is stored in memory and is processed during the next sequence. Concurrently, the sample reservoir is flushed and refilled with a fresh sample. At the conclusion of the scanning cycle, acid or standard may be added to the sample if required. Finally pump 1 is reactivated, valves 1 and 2 are switched to the A side, and the entire sequence is repeated following a delay to purge oxygen from the new sample.

Although this is a very brief description of the actual analysis procedure, the important point to consider is the flow-through nature of the electrodes and the relatively simple plumbing needed to automatically control the analysis. This plumbing can be altered or extended to provide more capability, such as the addition of auxiliary reservoirs, as indicated in Fig. 10, should the need arise.



LRO 4814-8-77B

Figure 9. Chemical cell for new flow-through ASV system.

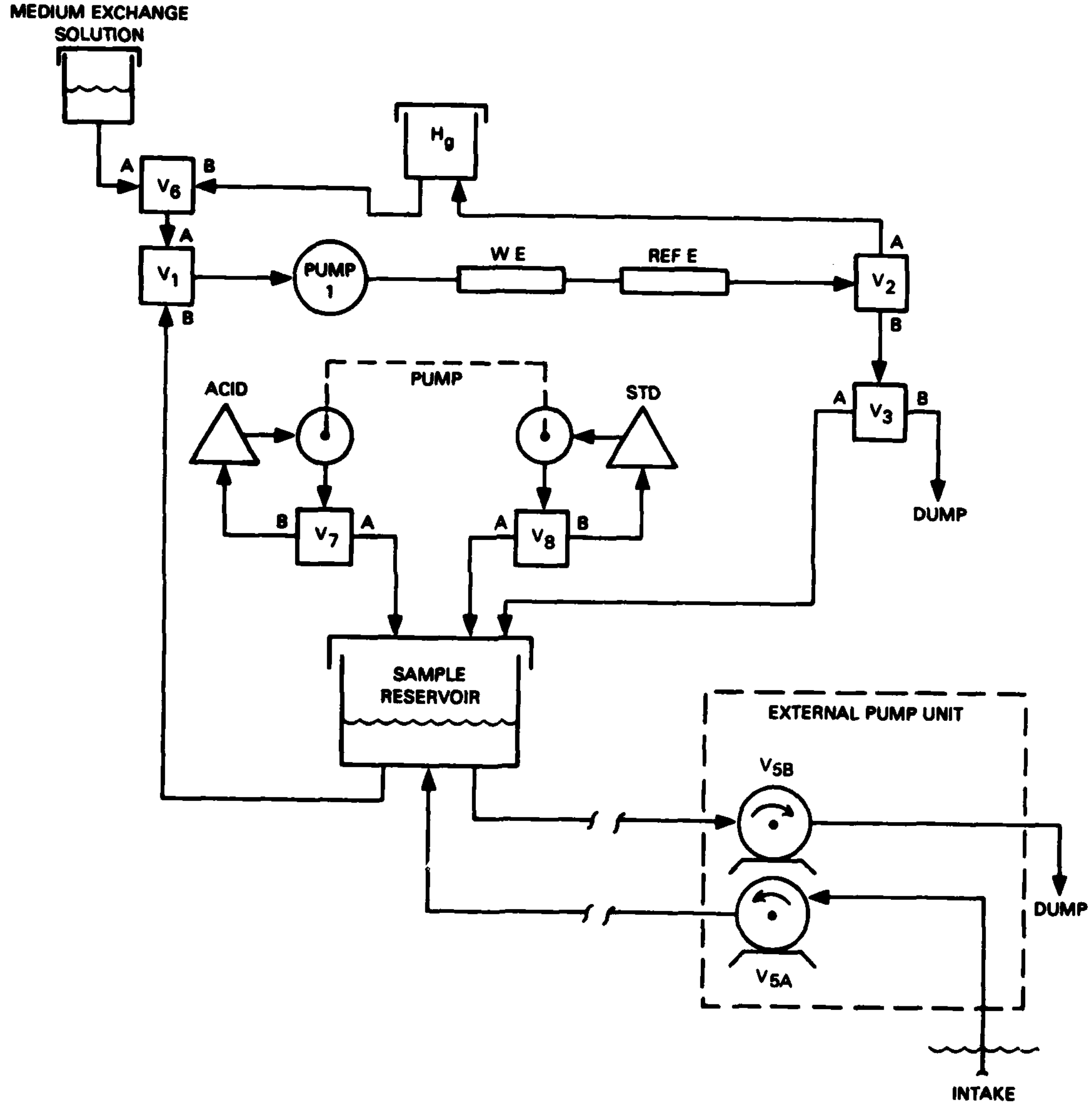


Figure 10. Chemical package plumbing schematic.

## SOFTWARE

The software developed for the microcomputer consists of two major system routines. The first is the controlling program for the ASV instrumentation. This program walks the operator through the initialization of the system by the use of the interactive questions and answers. The current program asks 28 questions covering all system parameter settings as well as the different operating mode options. Any of these inputs may be changed by the operator at any time during system operation. Once the instrument has been initialized, the start command is given by pushing the "GO" button and the analyses will proceed without further operator intervention. To allow system monitoring, critical parameters such as motor status, valve status, cycle status, and electrode voltage, which could affect an analysis, are continually displayed on the front panel via LEDs or digital panel meters.

The second major system program is the computing routine. This program operates on the stored digitized data, calculates the actual metal concentrations in ppb, and prints the results along with sample number, date, and sample acquisition time on the 40-column printer. The program operates on the digitized data generated during the potential scan. It automatically locates the current peaks for each metal and determines the validity of each peak by comparison with the programmed voltage range. The ranges are entered by the operator during initialization but can be changed at any time. Once the peaks have been located, their areas are calculated by the trapezoid rule for stepwise integration. The area values are now stored until the areas of the sample plus standard have also been calculated.

When the areas for both the samples and the sample plus standard have been determined, the program calculates the actual metal ion concentration by means of the following equation:

$$C_u = C_s \cdot \left( \frac{A_u}{A_{su} - A_u} \right)$$

Where

$C_u$ = Concentration of unknown metal ion in ppb

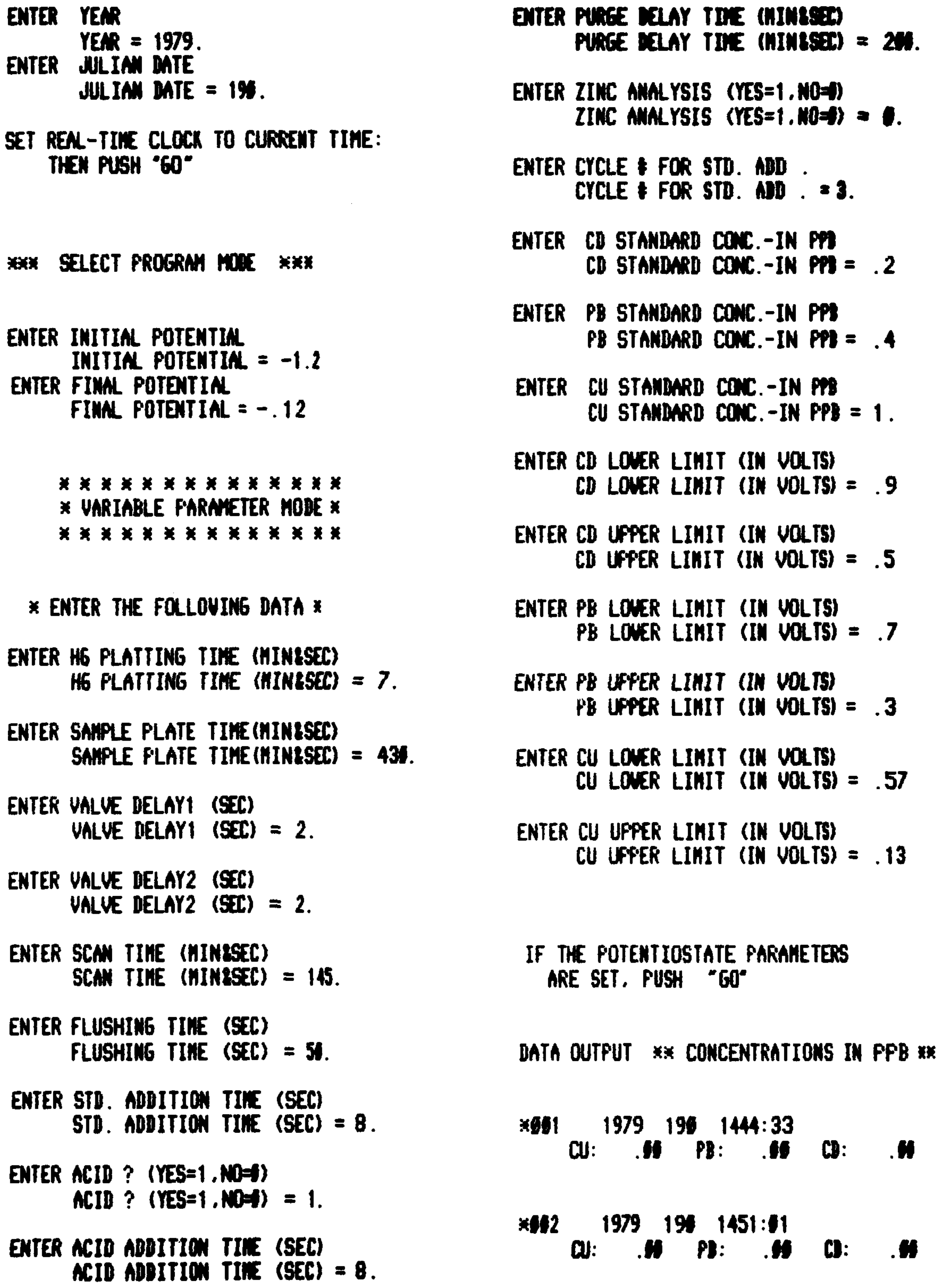
$C_s$ = Concentration of standard in ppb

$A_u$ = Peak area for metal ion

$A_{su}$ = Peak area for metal ion + standard addition

The results are then printed and the memory purged for the next set of data. A sample of the printout is shown in Fig. 11. A complete listing of all software is found in Appendix A.

As noted, the system's microcomputer module is accessible to the investigator and can be used for further program development. Provisions exist to allow the attachment of a CRT terminal and a dual floppy-disk mass-storage unit. When used in this fashion and with substitution of a simulator in place of the wet-chemistry electromechanical unit, the system becomes a powerful development tool, enabling new software and hardware designs to be developed and evaluated (including I/O simulation) without the need to invest in new equipment before a design is proven. The simulator also doubles as a system test and calibration



```
ENTER  YEAR
       YEAR = 1979.
ENTER  JULIAN DATE
       JULIAN DATE = 190.

SET REAL-TIME CLOCK TO CURRENT TIME:
    THEN PUSH "GO"


***  SELECT PROGRAM MODE  ***


ENTER INITIAL POTENTIAL
      INITIAL POTENTIAL = -1.2
 ENTER FINAL POTENTIAL
      FINAL POTENTIAL = -.12


     * * * * * * * * * * * * * *
     * VARIABLE PARAMETER MODE *
     * * * * * * * * * * * * * *


  * ENTER THE FOLLOWING DATA *

ENTER HG PLATTING TIME (MIN&SEC)
      HG PLATTING TIME (MIN&SEC) = 7.

ENTER SAMPLE PLATE TIME(MIN&SEC)
      SAMPLE PLATE TIME(MIN&SEC) = 430.

ENTER VALVE DELAY1  (SEC)
      VALVE DELAY1  (SEC) = 2.

ENTER VALVE DELAY2  (SEC)
      VALVE DELAY2  (SEC) = 2.

ENTER SCAN TIME  (MIN&SEC)
      SCAN TIME  (MIN&SEC) = 145.

ENTER FLUSHING TIME  (SEC)
      FLUSHING TIME  (SEC) = 50.

 ENTER STD. ADDITION TIME  (SEC)
      STD. ADDITION TIME  (SEC) = 8.

ENTER ACID ? (YES=1,NO=0)
      ACID ? (YES=1,NO=0) = 1.

ENTER ACID ADDITION TIME  (SEC)
      ACID ADDITION TIME  (SEC) = 8.
```

```
ENTER PURGE DELAY TIME (MIN&SEC)
      PURGE DELAY TIME (MIN&SEC) = 200.

ENTER ZINC ANALYSIS  (YES=1,NO=0)
      ZINC ANALYSIS  (YES=1,NO=0) = 0.

ENTER CYCLE # FOR STD. ADD .
      CYCLE # FOR STD. ADD . = 3.

ENTER  CD STANDARD CONC.-IN PPB
       CD STANDARD CONC.-IN PPB = .2

ENTER  PB STANDARD CONC.-IN PPB
       PB STANDARD CONC.-IN PPB = .4

ENTER  CU STANDARD CONC.-IN PPB
       CU STANDARD CONC.-IN PPB = 1.

ENTER CD LOWER LIMIT (IN VOLTS)
      CD LOWER LIMIT (IN VOLTS) = .9

ENTER CD UPPER LIMIT (IN VOLTS)
      CD UPPER LIMIT (IN VOLTS) = .5

ENTER PB LOWER LIMIT (IN VOLTS)
      PB LOWER LIMIT (IN VOLTS) = .7

ENTER PB UPPER LIMIT (IN VOLTS)
      PB UPPER LIMIT (IN VOLTS) = .3

ENTER CU LOWER LIMIT (IN VOLTS)
      CU LOWER LIMIT (IN VOLTS) = .57

ENTER CU UPPER LIMIT (IN VOLTS)
      CU UPPER LIMIT (IN VOLTS) = .13


 IF THE POTENTIOSTATE PARAMETERS
   ARE SET, PUSH  "GO"


DATA OUTPUT  ** CONCENTRATIONS IN PPB **


*001    1979  190  1444:33
     CU:    .00   PB:    .00   CD:    .00


*002    1979  190  1451:01
     CU:    .00   PB:    .00   CD:    .00
```

**Figure 11. Sample printout.**

unit for use in the field or laboratory, permitting a quick and safe determination of proper instrument function prior to actual use.

The ASV system's capacity for expansion or modification is limited only by the physical constraints, such as size and weight, that the designer might impose and by the total available memory, both core and mass storage. With the advent of bubble and CCD solid state mass-storage memories, it would be possible to combine very large amounts of memory in relatively small packages, facilitating further expansion of system capability with only a modest increase in size and weight.

## FUTURE DEVELOPMENTS

Work is currently underway to characterize a new electrode material that has shown great promise. The material is a low-temperature isotropic carbon in the form of a thin (1/16-in.) disk with a highly polished surface. These inexpensive disks can be used as received and are easily reconditioned by light polishing with fine diamond abrasive. Preliminary results indicate this electrode should function continuously for at least 24 hours before requiring reconditioning.

Investigations into what parameters affect sensitivity and a determination of the electrodes' actual useful lifetime will be carried out during the Verifront Equatorial Cruise of November 1979. This cruise will also provide the first extensive field test of the entire computer-controlled ASV system. Following the Verifront Project, the system will undergo continuous testing and evaluation in San Diego Bay throughout 1980, concurrent with the writing of a comprehensive operations manual.

These lengthy field tests will provide the necessary operation time to enable us to optimize the system's software and hardware configurations.

# REFERENCES

1. E. Barendrecht, Electroanalytical Chemistry, Marcel Dekker, New York, N.Y., Vol. 2, 1967.

2. W.D. Ellis, "Anodic Stripping Voltammetry," J. Chem. Ed., 50 A131-A147 (1973).

3. A. Zirino, S. Lieberman, and M.L. Healy. "Anodic Stripping Voltammetry of Trace Metals in Seawater," in Marine Electrochemistry (J.B. Berkowitz, R.A. Horne, M. Banus, P.L. Howard, M.J. Pryor and A.G. Whitnak, eds.), Electrochem. Soc., Princeton, N.J. 1973, pp. 319-332.

4. M. Whitfield, Chemical Oceanography, Academic Press, New York, N.Y., 1975.

5. S. Liberman, and A. Zirino. Anodic "Stripping Voltammetry of Zinc in Seawater with a Tubular Mercury-Graphite Electrode," Anal. Chem. 6, 20–23, (1974).

6. A. Zirino, S. Lieberman and C. Clavell, "Measurement of Copper and Zinc in San Diego Bay by Automated Anodic Stripping Voltammetry," Envir. Sci. Technol., 12, 73 (1978).

7. R. Fukai, and L. Huynh-Ngoc, "Direct Determination of Mercury in Seawater by Anodic Stripping Voltammetry with a Graphite Electrode," Anal. Chem. Acta., 83 375-379 (1975).

# INDEX TO APPENDIX
# A

## SYSTEM CONTROL AND COMPUTING PROGRAMS

All software was written in 6800 assembly language under Motorola's EDOS Operating system, using Motorola's Macro Assembler and Linking Loader.

Control Prog Size = ~8K    Computing Prog = ~4K

SECTION I: Utility Subroutines

SECTION II: Keyboard interface and A/D–D/A set-up subroutines

SECTION III: Main System Control Routines

SECTION IV: Computing Routine

# SECTION I:
## Utility Subroutines

```
001   BCDB    ***  BCD TO BINARY CONVERSION ROUTINE  ***

                        NAM    BCDB     VER.1.1     11-22-77    C
                   *
                   *  FILE NAMES:   &BCDB (S) / BCDB% (R) / BCDB (A)
                   *
                        OPT    REL
                   *
                        TTL    ***  BCD TO BINARY CONVERSION ROUTI
                   *
                   **************************************************
                   *
                   *    THIS SUBROUTINE CONVERTS 4 PACKED BCD DIGITS
                   *    16 BIT BINARY NUMBER.
                   *
                   *    INPUT IS IN A (MSB) & B (LSB) REG
                   *   OUTPUT IS IN A (MSB) & B (LSB)
                   **************************************************
                   *
                   *
0000 B7 0000   D BCDBIN STAA   SAVE1
0003 7F 0001   D        CLR    BINUPR
0006 17                 TBA
0007 C4 0F     A        ANDB   #$0F
0009 44                 LSRA
000A 44                 LSRA
000B 44                 LSRA
000C 44                 LSRA
000D 27 05 0014 TENLP   BEQ    DOHUND
000F CB 0A     A        ADDB   #10
0011 4A                 DECA
0012 20 F9 000D         BRA    TENLP
0014 0C          DOHUND CLC
0015 B6 0000   D        LDAA   SAVE1
0018 84 0F     A        ANDA   #$0F
001A 27 0A 0026 HUNLP   BEQ    DOTHOU
001C CB 64     A        ADDB   #100
001E 24 03 0023         BCC    HUNOO
0020 7C 0001   D        INC    BINUPR
0023 4A          HUNOO  DECA
0024 20 F4 001A         BRA    HUNLP
0026 B6 0000   D DOTHOU LDAA   SAVE1
0029 44                 LSRA
002A 44                 LSRA
002B 44                 LSRA
002C 44                 LSRA
002D B7 0000   D        STAA   SAVE1
0030 26 05 0037         BNE    THOU00
0032 B6 0001   D        LDAA   BINUPR
0035 20 0D 0044         BRA    XITBIN
0037 B6 0001   D THOU00 LDAA   BINUPR
003A 0C          THOULP CLC
003B CB E8     A        ADDB   #232
003D 89 03     A        ADCA   #$03
003F 7A 0000   D        DEC    SAVE1
0042 26 F6 003A         BNE    THOULP
0044 39          XITBIN RTS
                 *
                 *
                        XDEF   BCDBIN
```

```
002   BCDB   ***  BCD TO BINARY CONVERSION ROUTINE  ***
                       *
 0000                          DSCT
                       *
 0000      0001   A SAVE1   RMB     1
 0001      0001   A BINUPR  RMB     1
                       *
                               END
ERRORS 00000
```

```
                        NAM    RMULT
                 *
                 *  FILE NAMES:   &RMUL (S) / RMUL (R)
                 *
                        OPT    REL
                 *
                        TTL    ***  16 X 16 BIT MULTIPLY  ***
                 *
                 ************************************
                 *    UNSIGNED MULTIPLY ROUTINE
                 *
                 *   TO SET UP FOR USE:
                 *  LDAA  (HI BYTE)    16 BIT VALUE
                 *  LDAB  (LO BYTE)    MULTIPLICAN
                 *  INS
                 *  INS
                 *  INS
                 *  INS
                 *  INS
                 *  PSHB
                 *  PSHA
                 *  LDAA    (HI BYTE)  MULTIPLIER
                 *  LDAB  (LO BYTE)
                 *  PSHB
                 *  PSHA
                 *  DES
                 *  JSR MULT16
                 *
                 *   RETURNS WITH HI BYTE IN A REG
                 *     & LO BYTE IN B REG
                 *
                 ************************************
                 *
                 *
0000 30          MULT16 TSX
0001 86 10     A        LDAA   #16
0003 A7 02     A        STAA   2,X
0005 4F                 CLRA
0006 5F                 CLRB
0007 66 03     A        ROR    3,X
0009 66 04     A        ROR    4,X
000B 24 04 0011  NNEXT  BCC    RROTN
000D EB 06     A        ADDB   6,X
000F A9 05     A        ADCA   5,X
0011 46          RROTN  RORA
0012 56                 RORB
0013 66 03     A        ROR    3,X
0015 66 04     A        ROR    4,X
0017 6A 02     A        DEC    2,X
0019 26 F0 000B         BNE    NNEXT
001B 39                 RTS
                 *
                 *    EXTERNAL DEFINITIONS
                 *
                        XDEF   MULT16
                 *
                        END
ERRORS 00000
```

```
                         NAM     DV16     VER.1.1    11-22-77   CLAV
                  *
                  *  FILE NAMES:   &DV16 (S) / DV16 (R)
                  *
                         OPT     REL
                  *
                         TTL     ***  16 X 16 BIT DIVIDE ROUTINE  **
                  *
                  ******************************************************
                  *
                  *    THIS ROUTINE DIVDES TWO UNSIGNED 16 BIT
                  *    NUMBERS INTO EACH OTHER AND RETURNS A
                  *     16 BIT ANSWER.
                  *
                  *  TO USE:  ENTER WITH HI BYTE OF DIVIDEND IN A
                  *              & LO BYTE IN B
                  *           ADDRS. OF HI BYTE OF DIVISOR IN X
                  *
                  *      RETURNS:  A - HI BYTE RESULT
                  *                B - LO BYTE RESULT
                  *
                  ******************************************************
                  *
                  *
                  *
0000 37           DIV16  PSHB
0001 36                  PSHA
0002 A6 00      A        LDAA    X
0004 E6 01      A        LDAB    1,X
0006 37                  PSHB
0007 36                  PSHA
0008 34                  DES
0009 30                  TSX
000A 86 01      A        LDAA    #1
000C 6D 01      A        TST     1,X
000E 2B 0B 001B          BMI     DIV153
0010 4C           DIV151 INCA
0011 68 02      A        ASL     2,X
0013 69 01      A        ROL     1,X
0015 2B 04 001B          BMI     DIV153
0017 81 11      A        CMPA    #17
0019 26 F5 0010          BNE     DIV151
001B A7 00      A DIV153 STAA    X
001D A6 03      A        LDAA    3,X
001F E6 04      A        LDAB    4,X
0021 6F 03      A        CLR     3,X
0023 6F 04      A        CLR     4,X
0025 E0 02      A DIV163 SUBB    2,X
0027 A2 01      A        SBCA    1,X
0029 24 07 0032          BCC     DIV165
002B EB 02      A        ADDB    2,X
002D A9 01      A        ADCA    1,X
002F 0C                  CLC
0030 20 01 0033          BRA     DIV167
0032 0D           DIV165 SEC
0033 69 04      A DIV167 ROL     4,X
0035 69 03      A        ROL     3,X
0037 64 01      A        LSR     1,X
```

```
002   DV16   ***  16 X 16 BIT DIVIDE ROUTINE  ***

 0039 66 02     A        ROR    2,X
 003B 6A 00     A        DEC    X
 003D 26 E6 0025         BNE    DIV163
 003F 31                 INS
 0040 31                 INS
 0041 31                 INS
 0042 32                 PULA
 0043 33                 PULB
 0044 39                 RTS
                 *
                 *
                         XDEF   DIV16
                 *
                         END
ERRORS 00000
```

```
001   DIV3   ***  32 X 16 BIT DIVIDE  ***

                               NAM    DIV3     VER. 1.1   11-22-77   CLAV
                        *
                        *  FILE NAMES:    &DIV3 (S) / DIV3 (R)
                        *
                               OPT    REL
                        *
                               TTL    ***  32 X 16 BIT DIVIDE  ***
                        *
                        ***********************************************
                        *
                        *    THIS ROUTINE DIVIDES A 16 BIT NUMBER INTO A 32 BIT
                        *    NUMBER AND RETURNS A 16 BIT NUMBER.
                        *
                        ***********************************************
                        *
                        *
                        *
 0000 8D 4A 0040 DIV32  BSR    OVFTST
 0002 CE 0011  A L5     LDX    #17
 0005 5F         L0     CLRB
 0006 B6 0004  D        LDAA   DSORU
 0009 B1 0001  D        CMPA   DENDHU
 000C 27 30 003E        BEQ    L2
 000E 22 13 0023        BHI    L1
 0010 5C         L3     INCB
 0011 B6 0000  D        LDAA   DENDHL
 0014 B0 0005  D        SUBA   DSORL
 0017 B7 0000  D        STAA   DENDHL
 001A B6 0001  D        LDAA   DENDHU
 001D B2 0004  D        SBCA   DSORU
 0020 B7 0001  D        STAA   DENDHU
 0023 8C 0001  A L1     CPX    #1
 0026 27 20 0048        BEQ    L7
 0028 8D 39 0063        BSR    STSHF
 002A 79 0000  D        ROL    DENDHL
 002D 79 0001  D        ROL    DENDHU
 0030 FA 0003  D L6     ORAB   DENDLL
 0033 F7 0003  D        STAB   DENDLL
 0036 09                DEX
 0037 26 CC 0005        BNE    L0
 0039 FE 0002  D        LDX    DENDLU
 003C 0A                CLV
 003D 39                RTS
 003E B6 0005  D L2     LDAA   DSORL
 0041 B1 0000  D        CMPA   DENDHL
 0044 22 DD 0023        BHI    L1
 0046 20 C8 0010        BRA    L3
 0048 8D 19 0063 L7     BSR    STSHF
 004A 20 E4 0030        BRA    L6
 004C B6 0001  D OVFTST LDAA   DENDHU
 004F B1 0004  D        CMPA   DSORU
 0052 2E 0B 005F        BGT    OVFYES
 0054 2D 08 005E        BLT    OVFNO
 0056 B6 0005  D        LDAA   DSORL
 0059 B1 0000  D        CMPA   DENDHL
 005C 23 01 005F        BLS    OVFYES
 005E 39         OVFNO  RTS
 005F 32         OVFYES PULA
```

```
002   DIV3   ***  32 X 16 BIT DIVIDE  ***

 0060 32                    PULA
 0061 0B                    SEV
 0062 39                    RTS
 0063 78 0003   D STSHF     ASL     DENDLL
 0066 79 0002   D           ROL     DENDLU
 0069 39                    RTS
                  *
                  *
                            XDEF    DENDHU,DENDHL,DENDLU,DENDLL
                            XDEF    DSORU,DSORL,DIV32
                  *
 0000                       DSCT
                  *
 0000      0001   A DENDHL  RMB     1
 0001      0001   A DENDHU  RMB     1
 0002      0001   A DENDLU  RMB     1
 0003      0001   A DENDLL  RMB     1
 0004      0001   A DSORU   RMB     1
 0005      0001   A DSORL   RMB     1
                  *
                            END
ERRORS 00000
```

```
001   TIME    ***  REAL-TIME LED DISPLAY SUBROUTINE  ***
                          NAM    TIME     VER. 2   4-9-78     CLAVEL
                   *
                   *        FILE NAME:   &TIME (S)/ TIME (R)
                   *
                          OPT    REL
                   *
                          TTL    ***  REAL-TIME LED DISPLAY SUBROUTI
                   *
                   ***********************************************
                   *
                   *   THIS ROUTINE DISPLAYS THE CURRENT TIME ON THE
                   *   LED'S UNTIL THE "GO" BUTTON IS PUSHED.  TIME
                   *   APPROX. EVERY SEC..
                   *
                   *   THE ROUTINE IA A SUBROUTINE AND WILL ONLY BE
                   *   DURING INTIALIZATION TO ALLOW SETTING OF THE
                   *
                   *   "GO" COMES IN ON : CB1-PIA5
                   *
                   ***********************************************
                   *
 0000 B6 0000   A TIME    LDAA    P5BP      DUMMY READ TO CLR
                   *                       INTER. FLAGS
 0003 7C 0000   A         INC     TIMFLG    SET TIMFLG
 0006 BD 0000   A SKIP76  JSR     CLOCK     GET TIME AND DIPLAY
                   *
 0009 B6 0000   A         LDAA    P5BC      CHECK IF "GO" PUSHED
 000C 2B 02 0010          BMI     SKIP75    YES- RTS
                   *
 000E 20 F6 0006          BRA     SKIP76    NO- READ CLOCK AGAIN
                   *
 0010 7F 0000   A SKIP75  CLR     TIMFLG    CLR FLAG
 0013 39                  RTS               RETURN
                   *
                   *
                   *    ***  EXTERNAL REFERENCES  ***
                   *
                          XREF   TIMFLG,CLOCK,P5BC,P5BP
                   *
                   *
                   *    ***  EXTERNAL DEFINITIONS  ***
                   *
                          XDEF   TIME
                   *
                          END
ERRCRS 00000
```

```
001   PRINT1 ***  MESSAGE PRINTING SUBROUTINE ***
                                NAM    PRINT1   VER. 1    11-8-77    CLAVE
                         *
                         *  FILE NAMES:  &PRT1 (S) /PRT1 (R)
                         *
                                OPT    REL
                         *
                                TTL    ***  MESSAGE PRINTING SUBROUTINE **
                         *
                         *
 0005                           ORG    $0005
                         *
 0005      0002     A STRADR    RMB    2
                         *
 0000                           PSCT
                         *
 0000 DF 05         A PRINT1    STX    STRADR     MESS BUFF STARTING ADDR.
 0002 B6 0000       D           LDAA   BLOCK      # OF LINES IN MESS.
 0005 B7 0007       A           STAA   BLKADR
                         *
 0008 BD B2C3       A           JSR    SUB1       PRINT BUFF
 000B BD B2A0       A           JSR    SUB2
                         *
 000E 39                        RTS
                         *
                         *
                         *
                                XDEF   SUB1,SUB2,SUB3,BLOCK,BLKADR,STRADR,
                         *
           B2C3     A SUB1      EQU    $B2C3
           B2A0     A SUB2      EQU    $B2A0      DRIVER
           B38B     A SUB3      EQU    $B38B      LF
                         *
           0007     A BLKADR    EQU    $0007
                         *
 0000                           DSCT
                         *
 0000      0001     A BLOCK     RMB    1
                         *
                                END
ERRORS 00000
```

```
001   CLOCK  ***  REAL-TIME CLOCK INPUT SUBROUTINE  ***

                           NAM     CLOCK    VER. 3   4-18-79    CLAVEL
                     *
                     *      FILE:  &CLK (S)/ CLK (R)
                     *
                           OPT     REL
                     *
                           TTL     ***  REAL-TIME CLOCK INPUT SUBROUTI
                     *
                     *
                     ****************************************************
                     *     THIS SUBROUTINE READS IN THE CURRENT TIME
                     *     FROM THE REAL-TIME CLOCK IN BCD DIGITS (6),
                     *     ONE AT A TIME, BY TOGGLING THE CB2(PIA7)
                     *     LINE TO ADVANCE THE DIGIT INPUT.  THE
                     *     DIGITS ARE STORED IN (TBCDB1-6) AND ALSO
                     *     THE HR,MIN & SEC DIGITS ARE PACKED IN
                     *     TBCDH,TBCDM & TBCDS.
                     *
                     *     THIS ROUTINE IS ALSO USED BY THE RESET
                     *     PROG TO DISPLAY THE TIME ON THE LED'S TO
                     *     ALLOW SETTING THE CLOCK.
                     *
                     ****************************************************
                     *
                     *
 0000                      DSCT
                     *
 0000    0002   A DIGITU RMB     2
 0002    0001   A TBCDH  RMB     1
 0003    0001   A TBCDM  RMB     1
 0004    0001   A TBCDS  RMB     1
 0005    0001   A THRB   RMB     1
 0006    0002   A TBCDBU RMB     2
 0008    0001   A TSTORU RMB     1
 0009    0001   A TSTORL RMB     1
 000A    0001   A DISPCN RMB     1
                     *
                     *
 0200                      PSCT
                     *
                     *
 0000 0F           CLOCK   SEI                 SET INTERRUPT MASK
 0001 B6 0000  A           LDAA    P7BP        READ B SIDE OF PIA
 0004 43                   COMA                COMPLEMENT A
                     *                     BIT 7 SET?  STATR WITH SEC DI
 0005 2B 1C 0023           BMI     SKIP70   YES- GO TO SKIP70
                     *                     OTHERWISE TOGGLE CB2-PIA7
 0007 B6 0000  A           LDAA    P7BC
 000A 8B 08    A           ADDA    #$08        SET BIT 3
 000C B7 0000  A           STAA    P7BC
                     *
 000F C6 0F    A           LDAB    #15         100 US DELAY
 0011 5A           STALL   DECB
 0012 26 FD 0011           BNE     STALL
                     *
 0014 B6 0000  A           LDAA    P7BC
 0017 84 F7    A           ANDA    #$F7        CLR BIT 3
 0019 B7 0000  A           STAA    P7BC
```

```
                             *
  001C 86 C8      A                LDAA    #200
  001E 4A                STAL2     DECA                 1 MS DELAY
  001F 26 FD 001E                  BNE     STAL2
                             *
  0021 20 DD 0000                  BRA     CLOCK        START OVER
                             *
  0023 C6 06      A      SKIP70    LDAB    #6           SET UP COUNTER
  0025 CE 000B    D                LDX     #TBCDB1      LOAD X WITH 1ST BUFF ADDR.
  0028 84 0F      A      SKIP71    ANDA    #$0F         MASK OFF UPPER 4 BITS
                             *
  002A A7 00      A                STAA    X            PUT DIGIT INTO BUFF
                             *
  002C B6 0000    A                LDAA    P7BC          ON
  002F 8B 08      A                ADDA    #$08         TOGGLE CB2 FOR
  0031 B7 0000    A                STAA    P7BC         NEXT DIGIT
                             *
  0034 86 0F      A                LDAA    #15
  0036 4A                STAL1     DECA                 100μS
  0037 26 FD 0036                  BNE     STAL1        DELAY
                             *
  0039 B6 0000    A                LDAA    P7BC
  003C 84 F7      A                ANDA    #$F7         OFF CB2
  003E B7 0000    A                STAA    P7BC
                             *
  0041 5A                          DECB                 DONE 6 TIMES?
  0042 27 0C 0050                  BEQ     SKIP72       YES- GO TO SKIP72
                             *          NO- CONTINUE
  0044 08                          INX                  INC THE BUFF ADDR.
                             *
  0045 86 FA      A                LDAA    #250
  0047 4A                STAL3     DECA                 1.5 MS DELAY
  0048 26 FD 0047                  BNE     STAL3
                             *
  004A B6 0000    A                LDAA    P7BP         READ CLOCK
  004D 43                          COMA                 COMPLEMENT A
  004E 20 D8 0028                  BRA     SKIP71       LOOP
                             *
  0050 7D 0000    A      SKIP72    TST     TIMFLG       CALLED FROM TIME?
  0053 26 29 007E                  BNE     SKIP73       YES- GO TO SKIP73
                             *
                             *            OTHERWISE CONTINUE
  0055 B6 000C    D                LDAA    TBCDB2
  0058 48                          ASLA
  0059 48                          ASLA
  005A 48                          ASLA
  005B 48                          ASLA                        SEC'S DIGITS
  005C BB 000B    D                ADDA    TBCDB1      (PACK ~~LOWER 2~~
                             *                  ~~DIGITS~~ INTO TBCD~~L~~S
  005F B7 0004    D                STAA    TBCDS
                             *
  0062 B6 000E    D                LDAA    TBCDB4
  0065 48                          ASLA                        min's
  0066 48                          ASLA                 PACK ~~UPPER 2~~
  0067 48                          ASLA                 DIGITS INTO
  0068 48                          ASLA                 TBCD~~H~~M
  0069 BB 000D    D                ADDA    TBCDB3
                             *
```

```
006C B7 0003   D        STAA   TBCDM      MIN.'S
                   *
006F B6 0010   D        LDAA   TBCDB6
0072 48                 ASLA
0073 48                 ASLA              PACK HR'S
0074 48                 ASLA              INTO TBCDH
0075 48                 ASLA
0076 BB 000F   D        ADDA   TBCDB5
                   *
0079 B7 0002   D        STAA   TBCDH
                   *
007C 0E                 CLI
007D 39                 RTS               RETURN
                   *
                   *
007E 7F 0008   D SKIP73 CLR    TSTORU     CONVERT BCD DIGITS
0081 C6 06     A        LDAB   #6         TO SEG. CODE FOR
0083 CE 000B   D        LDX    #TBCDB1    DISPLAY
0086 FF 0006   D        STX    TBCDBU
0089 CE 0000   A        LDX    #DIGIT6
008C FF 0000   D        STX    DIGITU     TBCDB1 = 1'S SEC
                   *                   TBCDB6 = 10'S HR
                   *
008F FE 0006   D SEGCON LDX    TBCDBU
0092 A6 00     A        LDAA   X          TBCDB1 INTO A"
0094 B7 0009   D        STAA   TSTORL     "A" INTO TSTORL
                   *
0097 FE 0008   D        LDX    TSTORU     TBL + (OFSET) INTO "A
009A A6 00     A        LDAA   TBL,X
                   *
009C FE 0000   D        LDX    DIGITU     CODED DIGIT INTO DIGIT
009F A7 00     A        STAA   X
                   *
00A1 FE 0006   D        LDX    TBCDBU
00A4 08                 INX               INC BCD DIGIT BUFF
00A5 FF 0006   D        STX    TBCDBU
                   *
00A8 FE 0000   D        LDX    DIGITU     DEX SEG CODED
00AB 09                 DEX               DIGIT BUFF
00AC FF 0000   D        STX    DIGITU
                   *
00AF 5A                 DECB              IF NOT DONE 6
00B0 26 DD 008F         BNE    SEGCON     TIMES LOOP BACK
                   *
                   *
00B2 C6 3C     A        LDAB   #60        DISPLAY TIME FOR
00B4 F7 000A   D        STAB   DISPCN     900 MS ON LED'S
                   *
00B7 BD 0000   A SKIP74 JSR    DISPLA
                   *
00BA 7A 000A   D        DEC    DISPCN
00BD 26 F8 00B7         BNE    SKIP74
                   *
00BF 0E                 CLI
00C0 39                 RTS
                   *
                   *
                   *   *** EXTERNAL REFERENCES  ***
```

```
ØØ4   CLOCK  ***  REAL-TIME CLOCK INPUT SUBROUTINE  ***
                        *
                              XREF   DISPLA,TBL,DIGIT6,TIMFLG,P7BC,P7BP
                        *
                        *
                        *    ***  EXTERNAL DEFINITIONS  ***
                        *
                              XDEF   TBCDB1,TBCDB2,TBCDB3,TBCDB4.TBCDB5,
                              XDEF   TBCDH,TBCDM,TBCDS.CLOCK
                        *
                        *
 ØØØB                         DSCT
                        *
 ØØØB    ØØØ1   A TBCDB1 RMB     1
 ØØØC    ØØØ1   A TBCDB2 RMB     1
 ØØØD    ØØØ1   A TBCDB3 RMB     1
 ØØØE    ØØØ1   A TBCDB4 RMB     1
 ØØØF    ØØØ1   A TBCDB5 RMB     1
 ØØ1Ø    ØØØ1   A TBCDB6 RMB     1
                        *
                        *
                              END
ERRORS ØØØØØ
```

```
001   GENTM  ** GENERAL PURPOSE TIMING ROUTINE **
                          NAM    GENTM    VER. 1  12-15-77  CLAVELL
                   *
                   *  FILE NAMES*  &GNTM (S) / GENTM (R)
                   *
                          OPT    REL
                   *
                          TTL    ** GENERAL PURPOSE TIMING ROUTINE *
                   *
                   *******************************************
                   * TIME DELAY ROUTINE: TO USE LOAD A REG
                   * WITH 2 TIMES THE # OF SEC FOR DELAY
                   * THEN CALL GENTIM: IE. LDAA #12
                   * JSR GENTIM / WOULD GIVE A 6 SEC DELAY
                   *******************************************
                   *
                   *
 0000 CE F423  A GENTIM LDX    #$F423
 0003 09         DECRX  DEX
 0004 26 FD 0003        BNE    DECRX
 0006 CE 2A51  A        LDX    #$2A51
 0009 09         DECX   DEX
 000A 26 FD 0009        BNE    DECX
 000C 4A                DECA
 000D 26 F1 0000        BNE    GENTIM
                   *
 000F 39                RTS
                   *
                   *
                          XDEF   GENTIM
                   *
                          END
ERRORS 00000
```

THIS ROUTINE IS NOT PERFECTLY LINEAR:

| INPUT TIME (SEC) | ACTUAL TIME (SEC) |
|---|---|
| 5 | 5.2 |
| 10 | 10.1 |
| 15 | 15.0 |
| 20 | 19.9 |
| 30 | 29.8 |
| 40 | 39.6 |
| 60 | 59.2 |

## SECTION II:
## Keyboard Interface and A/D–D/A Set-Up Subroutines

```
                                 NAM    FRTQ     VER 14  9-19-79    CLAVELL
                         *
                         *  FILE NAMES:   &FRTQ (S) / FRTQ (R)
                         *
                                 OPT    REL
                         *
                                 TTL    ***  IP & FP INPUT ROUTINE  ***
                         *
                         **************************************************
                         *
                         * THIS SUBROUTINE ASKS FOR INPUTS FOR THE SCAN
                         * INITIAL AND FINAL POTENTIALS AND FORMS THE END
                         * POINT COMPARISON VALUE AND THE D/A INITIAL
                         * POTENTIAL VALUE.
                         *
                         **************************************************
                         *
                         *
                         *
                         *****      RANGE CALC. SUBROUTINE      ******
                         *
                         *  CALCULATE HIGH & LOW PEAK RANGE VALUES
                         *  FOR USE IN COMPT. STORE RESULTS INTO
                         *  LOWECO,LOWECO+1,ETC.
                         *  LOWECO HAS MOST NEG VALUE, (START OF PEAK)
                         *
                         *  EXAMPLE RANGE    ZN: -1.2 TO -.9
                         *  VALUES:          CD: -.87  TO -.5
                         *                   PB: -.65 TO -.25
                         *                   CU: - .55 TO -.15
                         *  VOLTAGE * 1000 = INTEGER VALUE (IV)
                         *  I.E.:  .65 = 650 ($028A) = (IV)  FOR PB LOW VA
                         *
                         *************************************************
                         *  (IP - IV) * 10/3 * 410/1000 =A/D BIT COUNTS
                         *************************************************
                         *
0000 8E 0F50  A RNGCAL LDS     #$0F50    RESET STACK PT
0003 B7 0002  D        STAA    TP
0006 F7 0003  D        STAB    TP+1
0009 B6 0018  D        LDAA    IPHI      CALCULATE (IP-IV) FOR
000C F6 0019  D        LDAB    IPLO      CD
000F F0 0003  D        SUBB    TP+1      LOW BYTE
0012 B2 0002  D        SBCA    TP        HI BYTE
                *
0015 31                INS
0016 31                INS
0017 31                INS
0018 31                INS
0019 31                INS
001A 37                PSHB
001B 36                PSHA              (IP-IV) * 10
001C 86 00    A        LDAA    #$00
001E C6 0A    A        LDAB    #$0A      MULTIPLIER ON STACK (* 10)
0020 37                PSHB
0021 36                PSHA
0022 34                DES
                *
```

```
0023 BD 0000  A          JSR    MULT16   16*16 MULTIPLY -32 BIT RES
                *
0026 B7 0000  A          STAA   DENDHU
0029 F7 0000  A          STAB   DENDHL   SET UP FOR 32 BIT DIVISION
002C EE 03    A          LDX    3,X
002E FF 0000  A          STX    DENDLU
0031 86 00    A          LDAA   #$00     DIVISOR (3)
0033 C6 03    A          LDAB   #$03
0035 B7 0000  A          STAA   DSORU
0038 F7 0000  A          STAB   DSORL    ((IP-IV) * 10/3)
                *
003B BD 0000  A          JSR    DIV32
                *
003E B6 0000  A          LDAA   DENDLU
0041 F6 0000  A          LDAB   DENDLL
                *
0044 31                  INS             ((IP-IV) *10/3) * 410
0045 31                  INS
0046 31                  INS
0047 31                  INS
0048 31                  INS
0049 37                  PSHB
004A 36                  PSHA
004B 86 01    A          LDAA   #$01     * 410
004D C6 9A    A          LDAB   #$9A
004F 37                  PSHB
0050 36                  PSHA
0051 34                  DES
                *
0052 BD 0000  A          JSR    MULT16
                *
0055 B7 0000  A          STAA   DENDHU
0058 F7 0000  A          STAB   DENDHL   ((IP-IV) * 10/3) *410/1000
005B EE 03    A          LDX    3,X
005D FF 0000  A          STX    DENDLU
                *
0060 86 03    A          LDAA   #$03     /1000
0062 C6 E8    A          LDAB   #$E8
0064 B7 0000  A          STAA   DSORU
0067 F7 0000  A          STAB   DSORL
                *
006A BD 0000  A          JSR    DIV32
                *
006D B6 0000  A          LDAA   DENDLU   A/D BIT COUNTS
0070 F6 0000  A          LDAB   DENDLL
                *
0073 8E 0F8E  A          LDS    #$0F8E   STACK BACK AS WAS
0076 39                  RTS
                *
                *
                *
                *
0077 BF 0000  D FRSTQ    STS    STK      SAVE STK PTR
007A 8E 0F50  A FRSTQ1   LDS    #$0F50   RESET STK PTR
                *
007D 7C 001E  D          INC    FRSTQF   FLAG TO INDICATE INTEG. OR
                *                        DEC. #:USED BY KBIN
0080 CE 0000  A          LDX    #MES30
```

```
003   FRTQ    ***  IP & FP INPUT ROUTINE  ***

 0083 BD 0000  A          JSR    PRINT
                   *
 0086 86 01    A          LDAA   #1
 0088 BD 0000  A          JSR    KBIN      INPUT IP
                   *
 008B B6 0000  A          LDAA   BINHI     CHK FOR SIGN BIT
 008E 2A 02 0092          BPL    ERROR1    IF NOT SET : ERROR
 0090 20 03 0095          BRA    A1
 0092 7E 026C  P ERROR1   JMP    ERROR
                   *
 0095 84 7F    A A1       ANDA   #$7F      MASK OFF SIGN BIT
                   *
 0097 B7 0018  D          STAA   IPHI
 009A F6 0000  A          LDAB   BINLO     STORE IP IN IPHI & LO
 009D F7 0019  D          STAB   IPLO      12 BIT VALUE
                   *
 00A0 CE 0000  A          LDX    #MES31    PRINT FP MESSAGE
 00A3 BD 0000  A          JSR    PRINT
                   *
 00A6 86 01    A          LDAA   #1
 00A8 BD 0000  A          JSR    KBIN
                   *
 00AB B6 0000  A          LDAA   BINHI
 00AE 2A 35 00E5          BPL    ADDFP     IF FP POS, GO TO ADDFP
 00B0 84 7F    A          ANDA   #$7F      OTHERWISE MASK SIGN BIT
                   *
 00B2 B7 0000  A          STAA   BINHI     STORE FP BACK TO BINHI
 00B5 B6 0018  D          LDAA   IPHI
 00B8 F6 0019  D          LDAB   IPLO
                   *
 00BB B1 0000  A          CMPA   BINHI     COMPARE IP TO FP
 00BE 27 07 00C7          BEQ    LOWCK
 00C0 2D 02 00C4          BLT    ERROR2
 00C2 20 0D 00D1          BRA    CONT      IS IP>FP?
 00C4 7E 026C  P ERROR2   JMP    ERROR
 00C7 F1 0000  A LOWCK    CMPB   BINLO     IF YES GO TO ERROR
 00CA 2D 02 00CE          BLT    ERROR3
 00CC 20 03 00D1          BRA    CONT
 00CE 7E 026C  P ERROR3   JMP    ERROR
                   *
 00D1 B6 0018  D CONT     LDAA   IPHI
 00D4 F6 0019  D          LDAB   IPLO
 00D7 F0 0000  A          SUBB   BINLO     DIFF=IP-FP
 00DA B2 0000  A          SBCA   BINHI
 00DD B7 0016  D          STAA   DIFHI
 00E0 F7 0017  D          STAB   DIFLO
 00E3 20 0F 00F4          BRA    DIFCK     GO TO DIFCK
                   *
 00E5 F6 0000  A ADDFP    LDAB   BINLO
 00E8 FB 0019  D          ADDB   IPLO      DIFF=IP+FP
 00EB B9 0018  D          ADCA   IPHI
                   *
 00EE B7 0016  D          STAA   DIFHI
 00F1 F7 0017  D          STAB   DIFLO     DIF IN DIFHI & LO
                   *
 00F4 81 0B    A DIFCK    CMPA   #$0B      IS DIFF > 3000 DEC.
 00F6 2E 04 00FC          BGT    ERROR4
 00F8 27 05 00FF          BEQ    DIFCK1    YES: ERROR
```

```
00FA 20 0C 0108           BRA    CONT1     NO: CONTINUE
00FC 7E 026C  P ERROR4  JMP    ERROR
00FF C1 B8    A DIFCK1  CMPB   #$B8
0101 2E 02 0105           BGT    ERROR5
0103 20 03 0108           BRA    CONT1
0105 7E 026C  P ERROR5  JMP    ERROR
                *
0108 CE 01FC  A CONT1   LDX    #ZERO     D/A ZERO VALUE
010B FF 0014  D         STX    ZERO1
010E 7E 01A4  P         JMP    SKIPZN
                *
                ************************************************
                *  CALCULATE PEAK RANGE VALUES ON X AXIS
                *  CALLED FROM "INIT"
                ************************************************
                *
0111 B6 0006  D RNG     LDAA   CD+2
0114 F6 0007  D         LDAB   CD+3
0117 BD 0000  P         JSR    RNGCAL    CD LOWER LIMIT
011A B7 0002  A         STAA   LOWECO+2
011D F7 0003  A         STAB   LOWECO+3
                *
0120 B6 0004  D         LDAA   CD
0123 F6 0005  D         LDAB   CD+1
0126 BD 0000  P         JSR    RNGCAL    CD UPPER LIMIT
0129 B7 0002  A         STAA   HIECO+2
012C F7 0003  A         STAB   HIECO+3
                *
012F B6 000A  D         LDAA   PB+2
0132 F6 000B  D         LDAB   PB+3
0135 BD 0000  P         JSR    RNGCAL    PB LOW
0138 B7 0004  A         STAA   LOWECO+4
013B F7 0005  A         STAB   LOWECO+5
                *
013E B6 0008  D         LDAA   PB
0141 F6 0009  D         LDAB   PB+1
0144 BD 0000  P         JSR    RNGCAL    PB HI
0147 B7 0004  A         STAA   HIECO+4
014A F7 0005  A         STAB   HIECO+5
                *
014D 7D 0000  A         TST    ZINK
0150 26 29 017B         BNE    ZNK
                *
0152 B6 000E  D         LDAA   CU+2
0155 F6 000F  D         LDAB   CU+3
0158 BD 0000  P         JSR    RNGCAL    CU LOW
015B B7 0006  A         STAA   LOWECO+6
015E F7 0007  A         STAB   LOWECO+7
                *
0161 B6 000C  D         LDAA   CU
0164 F6 000D  D         LDAB   CU+1
0167 BD 0000  P         JSR    RNGCAL    CU HI
016A B7 0006  A         STAA   HIECO+6
016D F7 0007  A         STAB   HIECO+7
                *
0170 7D 0000  A         TST    F2        CALLED FROM QFIX?
0173 27 03 0178         BEQ    J2        NO:RETURN INIT
0175 7E 0000  A         JMP    RTNPT     YES:RETURN TO QFIX
```

```
                *
0178 7E 0000  A J2      JMP    RENTR    RETURN TO INIT
                *
                *
017B B6 0012  D ZNK     LDAA   ZN+2
017E F6 0013  D         LDAB   ZN+3
0181 BD 0000  P         JSR    RNGCAL   ZN LOW
0184 B7 0000  A         STAA   LOWECO
0187 F7 0001  A         STAB   LOWECO+1
                *
018A B6 0010  D         LDAA   ZN
018D F6 0011  D         LDAB   ZN+1
0190 BD 0000  P         JSR    RNGCAL   ZN HI
0193 B7 0000  A         STAA   HIECO
0196 F7 0001  A         STAB   HIECO+1
                *
0199 7D 0000  A         TST    F2
019C 27 03 01A1         BEQ    J3
019E 7E 0000  A         JMP    RTNPT
                *
01A1 7E 0000  A J3      JMP    RENTR    RETURN TO INIT
                *
                *
                ************************************************
                *  SCALE IP FOR USE BY 10 BIT D/A
                ************************************************
                *
01A4 B6 0018  D SKIPZN  LDAA   IPHI
01A7 F6 0019  D         LDAB   IPLO     SCALE IP FOR D/A USE
01AA 31                 INS             (DEC. #) * 256/5000
01AB 31                 INS             RESULT IS THE # OF BITS
01AC 31                 INS             TO GIVE THE PROPER MICRO
01AD 31                 INS             AMP'S FROM THE D/A
01AE 31                 INS
01AF 37                 PSHB
01B0 36                 PSHA
01B1 86 01    A         LDAA   #$01
01B3 C6 00    A         LDAB   #$00
01B5 37                 PSHB
01B6 36                 PSHA            PUT MULTIPLIER ON STACK
01B7 34                 DES
                *
01B8 BD 0000  A         JSR    MULT16
                *
01BB B7 0000  A         STAA   DENDHU   SET UP FOR DIVISION
01BE F7 0000  A         STAB   DENDHL
01C1 EE 03    A         LDX    3,X
01C3 FF 0000  A         STX    DENDLU
                *
01C6 86 13    A         LDAA   #$13     LOAD DIVISOR
01C8 C6 88    A         LDAB   #$88
01CA B7 0000  A         STAA   DSORU    5000 DEC.
01CD F7 0000  A         STAB   DSORL
                *
01D0 BD 0000  A         JSR    DIV32
                *
01D3 B6 0000  A         LDAA   DENDLU   16 BIT ANSWER
01D6 F6 0000  A         LDAB   DENDLL
```

```
 01D9 B7 001A  D          STAA    IP10H     BIT VALUE
 01DC F7 001B  D          STAB    IP10L
               *
 01DF B6 0014  D          LDAA    ZERO1     ZERO VALUE
 01E2 F6 0015  D          LDAB    ZERO1+1
               *
 01E5 F0 001B  D          SUBB    IP10L
 01E8 B2 001A  D          SBCA    IP10H     SUB BIT VALUE
               *
 01EB F7 001B  D          STAB    IP10L     10 BIT D/A : OUTPUT
 01EE B7 001A  D          STAA    IP10H     INITIAL POTETIAL VALUE
               *
               *
               ********************************************
               **    SCALING FOR END POINT DETERMINATION
               **       (DIF * 10/3)
               ********************************************
               *
 01F1 B6 0016  D          LDAA    DIFHI
 01F4 F6 0017  D          LDAB    DIFLO     SCALE DIF BY 10/3
 01F7 31                  INS
 01F8 31                  INS
 01F9 31                  INS
 01FA 31                  INS
 01FB 31                  INS               SET UP STACK
 01FC 37                  PSHB
 01FD 36                  PSHA              MULTIPLICAN TO STACK
 01FE 86 00    A          LDAA    #$00
 0200 C6 0A    A          LDAB    #$0A
 0202 37                  PSHB
 0203 36                  PSHA              MULTIPLIER TO STACK
 0204 34                  DES
               *
 0205 BD 0000  A          JSR     MULT16
               *
 0208 B7 0000  A          STAA    DENDHU
 020B F7 0000  A          STAB    DENDHL    DIVIDEND
 020E EE 03    A          LDX     3,X
 0210 FF 0000  A          STX     DENDLU
               *
 0213 86 00    A          LDAA    #$00
 0215 C6 03    A          LDAB    #$03      DIVISOR
 0217 B7 0000  A          STAA    DSORU
 021A F7 0000  A          STAB    DSORL
               *
 021D BD 0000  A          JSR     DIV32
               *
 0220 B6 0000  A          LDAA    DENDLU    PUT ANSWER INTO A & B
 0223 F6 0000  A          LDAB    DENDLL
               *
               *
               ********************************************
               *     FORM COMPARISON VALUE FOR END OF SCAN
               **     SCALE DIF FOR 12 BITS
               **     (DIF * 0FFF/2710)
               *  (VOLT RANGE/1000 * 4095 BITS/10 VOLTS = # A/D
               ********************************************
               *
```

```
                          *
 0226 31                        INS
 0227 31                        INS
 0228 31                        INS
 0229 31                        INS
 022A 31                        INS
 022B 37                        PSHB
 022C 36                        PSHA                  MULTIPLICAN TO STACK
 022D 86 0F      A              LDAA    #$0F
 022F C6 FF      A              LDAB    #$FF
 0231 37                        PSHB
 0232 36                        PSHA                  MULTIPLIER TO STACK
 0233 34                        DES
                          *
 0234 BD 0000    A              JSR     MULT16
                          *
 0237 B7 0000    A              STAA    DENDHU
 023A F7 0000    A              STAB    DENDHL    DIVIDEND
 023D EE 03      A              LDX     3,X
 023F FF 0000    A              STX     DENDLU
                          *
 0242 86 27      A              LDAA    #$27
 0244 C6 10      A              LDAB    #$10      DIVISOR
 0246 B7 0000    A              STAA    DSORU
 0249 F7 0000    A              STAB    DSORL
                          *
 024C BD 0000    A              JSR     DIV32
                          *
 024F B6 0000    A              LDAA    DENDLU    16 BIT ANSWER INTO A & B
 0252 F6 0000    A              LDAB    DENDLL
 0255 B7 001C    D              STAA    FPCMVU    STORE A/D COMPARISON VALUE
 0258 F7 001D    D              STAB    FPCMVL
                          *
 025B 7F 001E    D              CLR     FRSTQF
 025E BE 0000    D              LDS     STK       RESET STAK PNTR
                          *
 0261 7D 0000    A              TST     F2        CALLED FROM QFIX?
 0264 27 03 0269                BEQ     J1        NO
 0266 7E 0000    A              JMP     RTNPT     YES: RETURN TO QFIX
                          *
 0269 7E 0000    A J1           JMP     RETINT
                          *
 026C CE 0000    A ERROR        LDX     #ERRMES
 026F 86 01      A              LDAA    #1
 0271 BD 0000    A              JSR     PRINT1
 0274 CE 0000    A              LDX     #ASCBUF
 0277 86 20      A              LDAA    #$20
 0279 A7 00      A              STAA    X
 027B 7E 007A    P              JMP     FRSTQ1
                          *
                          *
                          *
                                XREF    KBIN,PRINT,PRINT1,BINHI,BINLO,MES30
                                XREF    MULT16,DENDHU,DENDHL,DENDLU,DENDLL
                                XREF    DIV32,ASCBUF,RETINT,LOWECO,HIECO
                                XREF    ZINK,DSORU,DSORL,MES31,ERRMES,RENTR
                                XREF    F2,RTNPT
                          *
```

```
ØØ8   FRTQ    ***  IP & FP INPUT ROUTINE  ***

                         *
                         *
                                 XDEF    IPHI,IPLO,IP1ØH,IP1ØL,FPCMVU,FPCMVL
                                 XDEF    FRSTQF,FRSTQ,RNGCAL,CD,PB,CU,ZN,RNG
                         *
 ØØØØ                            DSCT
                         *
 ØØØØ      ØØØ2   A STK      RMB     2
 ØØØ2      ØØØ2   A TP       RMB     2
 ØØØ4      ØØØ4   A CD       RMB     4
 ØØØ8      ØØØ4   A PB       RMB     4
 ØØØC      ØØØ4   A CU       RMB     4
 ØØ1Ø      ØØØ4   A ZN       RMB     4
 ØØ14      ØØØ2   A ZERO1    RMB     2
 ØØ16      ØØØ1   A DIFHI    RMB     1
 ØØ17      ØØØ1   A DIFLO    RMB     1
 ØØ18      ØØØ1   A IPHI     RMB     1
 ØØ19      ØØØ1   A IPLO     RMB     1
 ØØ1A      ØØØ1   A IP1ØH    RMB     1
 ØØ1B      ØØØ1   A IP1ØL    RMB     1
 ØØ1C      ØØØ1   A FPCMVU   RMB     1
 ØØ1D      ØØØ1   A FPCMVL   RMB     1
 ØØ1E      ØØØ1   A FRSTQF   RMB     1
 ØØ1F      ØØØ2   A IP       RMB     2
 ØØ21      ØØØ1   A IPHI1    RMB     1
 ØØ22      ØØØ1   A IPLO1    RMB     1
                         *
           Ø1FC   A ZERO     EQU     $Ø1FC    ADJUSTED TO GIVE PROPER
                         *                         READ OUT ON PANEL METER
                         *               .1 UA (D/A OUTPUT)/ MV (PANEL READ
                         *
                         *
                                 END
ERRORS ØØØØØ
```

```
                             NAM    KBIN     VER. 5   8-2-79    CIAVELL
                       *
                       *  FILE NAMES:  &KBIN (S) /KBIN (R)
                       *
                             OPT    REL
                       *
                             TTL    ***  KEY-BOARD INPUT SUBROUTINE  **
                       *
                       ***********************************************
                       *
                       *    THIS SUBROUTINE INPUTS DATA FROM A 16 KEY
                       *    KEYBOARD AND DISPLAYS IT ON LED'S AND STORES
                       *    IT FOR USE BY OTHER ROUTINES.
                       *
                       ***********************************************
                       *
                       *
                       *
 0000                        BSCT
                       *
 0000      0001      A BLANK  RMB    1
 0001      0006      A ASCNUM RMB    6
                       *
 0000                        PSCT
                       *
 0000      03        A TTBL  FCB    $03,$9F,$25,$0D,$99,$49,$41,$1F,$01
 0001      9F        A
 0002      25        A
 0003      0D        A
 0004      99        A
 0005      49        A
 0006      41        A
 0007      1F        A
 0008      01        A
 0009      09        A       FCB    $09,$11,$C0,$63,$85,$61,$71
 000A      11        A
 000B      C0        A
 000C      63        A
 000D      85        A
 000E      61        A
 000F      71        A
                       *
                       *
                       *
 0010 B7 0004  D KBIN     STAA   MAXDIG    STORE SCALE FACTOR
 0013 7F 0012  D          CLR    BINLO
 0016 7F 0001  D CLEAR    CLR    CHRCNT
 0019 7F 0000  D          CLR    DECFLG
                       *
 001C 7F 0003  D          CLR    BCDCNT
 001F 7F 0002  D          CLR    NUMCNT
 0022 7F 0011  D          CLR    BINHI
 0025 CE FFFF  A          LDX    #$FFFF    LOAD DIGIT1&2 WI
 0028 FF 000B  D          STX    DIGIT1    BLANKS
 002B CE EFEF  A          LDX    #$EFEF    FILL DIGIT3-6 WI
 002E FF 000D  D          STX    DIGIT3    UNDERLINES
 0031 FF 000F  D          STX    DIGIT5
 0034 CE 0000  A          LDX    #0
```

```
0037 FF 0013  D          STX    BCDHI     CLEAR BCD BUFFER
003A CE 2020  A          LDX    #$2020
003D DF 01    B          STX    ASCNUM    CLEAR ASCII BUFFER
003F DF 03    B          STX    ASCNUM+2
0041 DF 05    B          STX    ASCNUM+4
0043 7C 0012  D          INC    BINLO     TEST TO SEE IF WE
0046 28 08 0050          BVC    LIGHTS    HAVE RET FROM AN ERROR
0048 7F 0011  D          CLR    BINHI     IF SO,
004B 7F 0012  D          CLR    BINLO
004E 20 10 0060          BRA    BEGIN     GO TO BEGIN
                *
0050 BD 0000  A LIGHTS   JSR    DISPLA    DISPLAY LED'S
0053 B6 0000  A          LDAA   P7AP      READ KEY-BOARD
0056 84 1F    A          ANDA   #$1F      MASK UPPER 3 BITS
0058 27 F6 0050          BEQ    LIGHTS    NO DATA-LOOP BACK
                *
005A F6 0012  D          LDAB   BINLO     TEST TO SEE IF WE
005D 5C                  INCB             HAVE RET FROM AN ERROR
005E 29 B6 0016          BVS    CLEAR     IF SO, GO TO CLEAR
0060 7C 0001  D BEGIN    INC    CHRCNT    INCREMENT CHRCNT
0063 84 0F    A          ANDA   #@17      MASK OFF UPPER 4 BITS
0065 81 09    A          CMPA   #@11      A BETWEEN 0 &9?
0067 2F 1C 0085          BLE    NUMBER    IF SO, GO TO NUMBER
0069 81 0D    A          CMPA   #@15      ELSE, A=-?
006B 26 03 0070          BNE    JUMP1     IF SO,GO TO MINUS
006D 7E 0115  P          JMP    MINUS
0070 81 0E    A JUMP1    CMPA   #@16      ELSE, A=.?
0072 26 03 0077          BNE    JUMP2     IF SO,GO TO DECML
0074 7E 012F  P          JMP    DECML
0077 81 0B    A JUMP2    CMPA   #@13      ELSE, A=CHG?
0079 27 9B 0016          BEQ    CLEAR     IF SO, GO TO CLEAR
007B 81 0A    A          CMPA   #@12      ELSE, A=ENT?
007D 26 03 0082          BNE    JUMP3     IF SO, GO TO ENTER
007F 7E 0170  P          JMP    ENTER
0082 7E 01F6  P JUMP3    JMP    UGOOF
                *
0085 7C 0002  D NUMBER   INC    NUMCNT    INCREMENT NUMBER COUNT
0088 7C 0003  D          INC    BCDCNT    INCREMENT BCD COUNT
008B C6 04    A          LDAB   #4        BCDCNT <=4
008D F1 0003  D          CMPB   BCDCNT
0090 2D 02 0094          BLT    UGOF1     IF NOT, GO TO UGOOF
0092 20 03 0097          BRA    JUMP5
0094 7E 01F6  P UGOF1    JMP    UGOOF
0097 B7 0006  D JUMP5    STAA   TADRL     STORE A IN TADRL
009A B6 0011  D          LDAA   BINHI     LOAD A WI ACTUAL # OF DIGI
                *           BEFORE DECIMAL PT.
009D 84 7F    A          ANDA   #$7F      MASK OFF SIGN BIT
009F B1 0004  D          CMPA   MAXDIG    TOO MANY DIGITS ENTERED?
00A2 2F 03 00A7          BLE    OKNUM
00A4 7E 01F6  P          JMP    UGOOF
                *
00A7 B6 0006  D OKNUM    LDAA   TADRL     ELSE, PUT CHAR IN A
00AA F6 0003  D          LDAB   BCDCNT    1ST NUMBER?
00AD C1 01    A          CMPB   #1
00AF 27 0B 00BC          BEQ    BCDST     IF SO, GO TO BCDST
00B1 C6 04    A          LDAB   #4        ELSE, SHIFT BCD BUFFER
00B3 78 0014  D BCDL     ASL    BCDLO     LEFT 4 BITS
00B6 79 0013  D          ROL    BCDHI
```

```
 00B9 5A                  DECB
 00BA 2E F7 00B3          BGT    BCDL
 00BC BB 0014  D BCDST    ADDA   BCDLO     STORE # IN 4 LEAST SIGN
 00BF B7 0014  D          STAA   BCDLO     BITS OF BCDLO
 00C2 B6 0002  D          LDAA   NUMCNT
 00C5 81 01    A          CMPA   #1
 00C7 27 1B 00E4          BEQ    LEDST     IF SO, GO TO LEDST
 00C9 CE 0010  D          LDX    #DIGIT6   IF NOT, CAL # TIMES
 00CC 80 02    A          SUBA   #2        TO SHIFT LED BUFFER
 00CE 27 04 00D4          BEQ    SECNUM
 00D0 09         LEDL1    DEX
 00D1 4A                  DECA
 00D2 26 FC 00D0          BNE    LEDL1     SHIFT LED BUFFER
 00D4 B6 0002  D SECNUM   LDAA   NUMCNT    UNTIL ALL NUMBERS
 00D7 4A                  DECA             HAVE BEEN LEFTSHIFTED
 00D8 E6 00    A LEDL2    LDAB   X         ONE LOCATION
 00DA 09                  DEX
 00DB E7 00    A          STAB   X
 00DD 4A                  DECA
 00DE 27 04 00E4          BEQ    LEDST
 00E0 08                  INX
 00E1 08                  INX
 00E2 20 F4 00D8          BRA    LEDL2
 00E4 C6 00    A LEDST    LDAB   #$00
 00E6 F7 0005  D          STAB   TADRU
 00E9 FE 0005  D          LDX    TADRU     STORE NEWEST CHAR
 00EC E6 00    A          LDAB   TBL,X     IN LED BUFFER
 00EE F7 0010  D          STAB   DIGIT6
 00F1 B6 0006  D          LDAA   TADRL     PUT CHAR IN A
 00F4 8B 30    A          ADDA   #@60      CONVERT IT TO ASCII
 00F6 CE 0001  B ASCST    LDX    #ASCNUM
 00F9 FF 0007  D          STX    TASCII
 00FC F6 0007  D          LDAB   TASCII
 00FF F7 0005  D          STAB   TADRU     STORE IN ASCII BUFFER
 0102 F6 0001  D          LDAB   CHRCNT
 0105 F7 0006  D          STAB   TADRL
 0108 FE 0005  D          LDX    TADRU
 010B A7 00    B          STAA   ASCNUM-1,X
 010D 86 01    A          LDAA   #1
 010F BD 0000  A          JSR    GENTIM    KEY IN DELAY
                 *
 0112 7E 0050  P          JMP    LIGHTS
                 *
 0115 F6 0001  D MINUS    LDAB   CHRCNT    IS - THE FIRST CHAR?
 0118 C1 01    A          CMPB   #1
 011A 26 02 011E          BNE    UGOF5     IF NOT, GOT TO UGOOF
 011C 20 03 0121          BRA    JUMPA
 011E 7E 01F6  P UGOF5    JMP    UGOOF
 0121 86 FD    A JUMPA    LDAA   #$FD      IF SO, PUT - IN DIGIT1
 0123 B7 000B  D          STAA   DIGIT1
 0126 86 80    A          LDAA   #$80      SET MSB OF BINHI
 0128 B7 0011  D          STAA   BINHI
 012B 86 2D    A          LDAA   #@55      SEND OUT ASCII MINUS TO
 012D 20 C7 00F6          BRA    ASCST     ASCII BUFFER
                 *
 012F 86 01    A DECML    LDAA   #1
 0131 B1 0001  D          CMPA   CHRCNT
 0134 26 05 013B          BNE    NF
```

```
0136 B7 0002  D        STAA   NUMCNT
0139 20 1C 0157        BRA    DECST
013B F6 0000  D NF     LDAB   DECFL3
013E 5D                TSTB
013F 26 02 0143        BNE    UGOF4
0141 20 03 0146        BRA    JUMP8
0143 7E 01F6  P UGOF4  JMP    UGOOF
0146 F6 000B  D JUMP3  LDAB   DIGIT1
0149 C1 FD    A        CMPB   #$FD
014B 26 0A 0157        BNE    DECST
014D C6 02    A        LDAB   #2
014F F1 0001  D        CMPB   CHRCNT
0152 26 03 0157        BNE    DECST
0154 B7 0002  D        STAA   NUMCNT
0157 F6 0010  D DECST  LDAB   DIGITS
015A C4 FE    A        ANDB   #$FE
015C F7 0010  D        STAB   DIGITS
015F B6 0011  D        LDAA   BINHI
0162 BB 0003  D        ADDA   BCDCNT
0165 B7 0011  D        STAA   BINHI
0168 86 2E    A        LDAA   #$2E
016A 7C 0000  D        INC    DECFLG
016D 7E 00F6  P        JMP    ASCST
                *
0170 7D 0000  D ENTER  TST    DECFLG   IS DECFLG SET?
0173 26 03 0178        BNE    OK1      YES-GO TO OK1
0175 7E 01F6  P        JMP    UGOOF    NO-DEC.PT. NOT ENTERED-ERR
0178 B6 0000  A OK1    LDAA   FRSTQF   FLAG SET?
017B 27 53 01D0        BEQ    INTINP   NO GO TO INTEGER INPUT
                *
                *    DECIMAL  INPUT  SECTION
                *
017D F6 0011  D        LDAB   BINHI    # DIG BEFORE DEC. PT.
0180 C4 7F    A        ANDB   #$7F     MASK SIGN BIT
                *
0182 27 25 01A9        BEQ    ZERO1    IF NO DIG. GO TO ZERO1
                *
0184 B6 0003  D        LDAA   BCDCNT
0187 81 04    A        CMPA   #4       4 #'S ENTERED?
0189 27 52 01DD        BEQ    ASCPR    YES- GO TO ASCPR
                *
018B 81 03    A        CMPA   #3
018D 27 12 01A1        BEQ    ZERO2    3 #'S/PACK WITH 1-0
                *
018F 81 02    A        CMPA   #2
0191 27 12 01A5        BEQ    ZERO3    2 #'S/PACK WITH 2-0'S
                *
0193 86 0C    A        LDAA   #12      OTHERWISE PACK WIHT 3-0'S
0195 78 0014  D BCDPAK ASL    BCDLO
0198 79 0013  D        ROL    BCDHI    DO 12 SHIFTS
019B 4A                DECA
019C 26 F7 0195        BNE    BCDPAK
                *
019E 7E 01DD  P        JMP    ASCPR
                *
01A1 86 04    A ZERO2  LDAA   #4       PACK WITH 1-0
01A3 20 F0 0195        BRA    BCDPAK
                *
```

```
 01A5 86 08     A ZERO3  LDAA    #8        PACK WITH 2-0'S
 01A7 20 EC 0195           BRA     BCDPAK
                  *
 01A9 F6 0003   D ZERO1  LDAB    BCDCNT    4 #S ENTERED?
 01AC C1 04     A        CMPB    #4        YES=ERROR
 01AE 27 02 01B2         BEQ     HOP
 01B0 20 03 01B5         BRA     HOP1
 01B2 7E 01F6   P HOP    JMP     UGOOF
                  *
 01B5 C1 03     A HOP1   CMPB    #3
 01B7 27 24 01DD         BEQ     ASCPR
                  *
 01B9 C1 02     A        CMPB    #2        PACK WITH 1-0
 01BB 26 05 01C2         BNE     ZERO4
 01BD 86 04     A        LDAA    #4
 01BF 7E 0195   P        JMP     BCDPAK
                  *
 01C2 C1 01     A ZERO4  CMPB    #1        1 # ENTERED?
 01C4 26 02 01C8         BNE     HOP2      NO- GO TO UGOOF
 01C6 20 03 01CB         BRA     HOP3
                  *
 01C8 7E 01F6   P HOP2   JMP     UGOOF
                  *
 01CB 86 08     A HOP3   LDAA    #8        YES-PACK WITH 2-0'S
 01CD 7E 0195   P        JMP     BCDPAK
                  *
                  *
                  *    INTEGER INPUT SECTION
                  *
                  *
 01D0 F6 0011   D INTINP LDAB    BINHI     # OF DIG BEFORE DEC PT
 01D3 C4 7F     A        ANDB    #$7F      MASK SIGN BIT
                  *
 01D5 F1 0003   D        CMPB    BCDCNT    CMP # DIG BEFORE DEC PT
 01D8 27 03 01DD         BEQ     ASCPR     TO TOTAL # OF DIGITS,IF =
 01DA 7E 01F6   P        JMP     UGOOF
                  *
 01DD BD 0000   A ASCPR  JSR     PRINT     PRINT ASCII BFFER
 01E0 B6 0013   D        LDAA    BCDHI
 01E3 F6 0014   D        LDAB    BCDLO
 01E6 BD 0000   A        JSR     BCDBIN    CONVERT BCD TO BINARY
 01E9 F7 0012   D        STAB    BINLO
 01EC F6 0011   D        LDAB    BINHI     STORE BINARY RESULT
 01EF C4 80     A        ANDB    #$80      SAVING MSB OF BINHI
 01F1 1B                 ABA               IF SET
 01F2 B7 0011   D        STAA    BINHI
 01F5 39                 RTS
                  *
 01F6 CE 83FD   A UGOOF  LDX     #$83FD    WRITE OUT U-GOOF
 01F9 FF 000B   D        STX     DIGIT1    U-
 01FC CE 4103   A        LDX     #$4103
 01FF FF 000D   D        STX     DIGIT3    GO
 0202 CE 0371   A        LDX     #$0371
 0205 FF 000F   D        STX     DIGIT5    OF
 0208 86 7F     A        LDAA    #$7F
 020A B7 0012   D        STAA    BINLO
 020D 7E 0050   P        JMP     LIGHTS
                  *
```

```
006   KBIN   ***  KEY-BOARD INPUT SUBROUTINE  ***

                         *
                                 XREF    DISPLA,PRINT,FRSTQ,BCDBIN
                                 XREF    TBL,FRSTQF,P7AP,GENTIM
                         *
                         *
                         *
                                 XDEF    BINHI,BINLO,BCDHI,BCDLO,DIGIT0,DIGI
                                 XDEF    DIGIT3,DIGIT4,DIGIT5,DIGIT6,ASCNUM,
                                 XDEF    TADRU,TADRL,TTBL,KBIN
                         *
  0000                           DSCT
                         *
  0000     0001    A  DECFLG  RMB      1
  0001     0001    A  CHRCNT  RMB      1
  0002     0001    A  NUMCNT  RMB      1
  0003     0001    A  BCDCNT  RMB      1
  0004     0001    A  MAXDIG  RMB      1
  0005     0001    A  TADRU   RMB      1
  0006     0001    A  TADRL   RMB      1
  0007     0002    A  TASCII  RMB      2
  0009     0001    A  COUNT   RMB      1
  000A     0001    A  DIGIT0  RMB      1
  000B     0001    A  DIGIT1  RMB      1
  000C     0001    A  DIGIT2  RMB      1
  000D     0001    A  DIGIT3  RMB      1
  000E     0001    A  DIGIT4  RMB      1
  000F     0001    A  DIGIT5  RMB      1
  0010     0001    A  DIGIT5  RMB      1
  0011     0001    A  BINHI   RMB      1
  0012     0001    A  BINLO   RMB      1
  0013     0001    A  BCDHI   RMB      1
  0014     0001    A  BCDLO   RMB      1
                         *
                         *
                                 END
ERRORS 00000
```

```
                           NAM     PRNT     VER 2    4-20-78    CLAVEL
                     *
                     *  FILE NAMES:  &PRNT (S) / PRNT (R)
                     *
                           OPT     REL
                     *
                           TTL     ***  MAIN PRINTING SUBROUTINE  ***
                     *
                     **********************************************
                     *
                     *    THIS SUBROUTINE FORMATS THE INPUT & OUTPUT
                     *    PARAMETER MESSAGES AND DOES THE PRINTING.
                     *
                     **********************************************
                     *
                     *
0000                       BSCT
                     *
0000     0028   A ASCBUF RMB     40
                     *
                     *
0000                       PSCT
                     *
0000 FF 0000   D PRINT  STX     TMESU     SAVE ADR OF MESSAGE
0003 CE 0000   B        LDX     #ASCBUF   PUT ADR OF BUF IN TADRU
0006 FF 0000   A        STX     TADRU
0009 B6 0000   A        LDAA    TADRL
000C D6 00     B        LDAB    ASCBUF    CALLED FROM KBIN?
000E C1 45     A        CMPB    #$45      IF SO, CLEAR 1ST FIVE LOC
0010 27 04 0016         BEQ     FIVE
0012 C6 28     A        LDAB    #$28
0014 20 02 0018         BRA     SETLIM    IF NOT, CLEAR ENTIRE 40
0016 C6 05     A FIVE   LDAB    #5        WORD BUFFER
0018 1B          SETLIM ABA
0019 B7 0000   A        STAA    TADRL
001C 86 20     A        LDAA    #$20      FILL BUFFER W/ BLANKS
001E A7 00     A BLANKS STAA    X
0020 08                 INX
0021 BC 0000   A        CPX     TADRU
0024 26 F8 001E         BNE     BLANKS
0026 C1 05     A        CMPB    #5        CALLED FROM KBIN?
0028 27 46 0070         BEQ     REPRNT    IF SO, GO TO REPRNT
002A 86 06     A        LDAA    #6
002C CE 0000   B        LDX     #ASCBUF   FILL ASCII PRINT BUF
002F C6 45     A        LDAB    #$45      W/ E
0031 E7 00     A        STAB    X
0033 08                 INX
0034 C6 4E     A        LDAB    #$4E      N
0036 E7 00     A        STAB    X
0038 08                 INX
0039 C6 54     A        LDAB    #$54      T
003B E7 00     A        STAB    X
003D 08                 INX
003E C6 45     A        LDAB    #$45      E
0040 E7 00     A        STAB    X
0042 08                 INX
0043 C6 52     A        LDAB    #$52      R
0045 E7 00     A        STAB    X
```

```
0047 08                       INX
0048 08                       INX
0049 FF 0000  A               STX     TADRU
004C FE 0000  D FILBUF        LDX     TMESU     TRANSFER MESSAGE BUFF
004F E6 00    A               LDAB    X         TO PRINT BUFFER
0051 4C                       INCA              UNTIL COLON IS REACHED
0052 C1 3A    A               CMPB    #$3A
0054 27 15 006B               BEQ     COLON     COLON FOUND?
0056 FE 0000  A               LDX     TADRU     IF NO, CONTINUE
0059 E7 00    A               STAB    X
005B 7C 0000  A               INC     TADRL
005E 7C 0001  D               INC     TMESL
0061 7D 0001  D               TST     TMESL
0064 26 03 0069               BNE     SK10
0066 7C 0000  D               INC     TMESU
0069 20 E1 004C SK10          BRA     FILBUF
                *
006B B7 0002  D COLON         STAA    COLLOC    SAVE COLON LOC
006E 20 30 00A0               BRA     WRITB     GO TO PRINT BUFFER
0070 B6 0002  D REPRNT        LDAA    COLLOC
0073 B7 0000  A               STAA    TADRL     POINT TO LOC OF COLON
0076 C6 3D    A               LDAB    #$3D      PUT IN = SIGN
0078 FE 0000  A               LDX     TADRU
007B E7 00    B               STAB    ASCBUF,X
007D 8B 02    A               ADDA    #2
007F B7 0000  A               STAA    TADRL
0082 CE 0000  A               LDX     #ASCNUM   GET ADR OF ASCII #
0085 FF 0000  D               STX     TMESU
0088 F6 0000  A               LDAB    CHRCNT
008B FE 0000  D XFER          LDX     TMESU     TRANSFER ASCII #
008E A6 00    A               LDAA    X         TO ASCBUF
                *
0090 FE 0000  A               LDX     TADRU
0093 A7 00    B               STAA    ASCBUF,X
0095 5A                       DECB
0096 27 08 00A0               BEQ     WRITB
0098 7C 0001  D               INC     TMESL
009B 7C 0000  A               INC     TADRL
009E 20 EB 008B               BRA     XFER
00A0 CE 0000  B WRITB         LDX     #ASCBUF   PRINT OUT ENTIRE BUFF
00A3 DF 05    A               STX     $05
00A5 86 01    A               LDAA    #1
00A7 97 07    A               STAA    $07
00A9 BD B2C3  A               JSR     $B2C3
00AC BD B2A0  A               JSR     $B2A0
00AF 39                       RTS
                *
                *
                              XREF    TADRU,TADRL,CHRCNT,ASCNUM
                *
                *
                              XDEF    COLLOC,ASCBUF,PRINT
                *
0000                          DSCT
                *
                *
0000      0001  A TMESU       RMB     1
0001      0001  A TMESL       RMB     1
```

```
ØØ3   PRNT   ***  MAIN PRINTING SUBROUTINE   ***

 ØØØ2   ØØØ1  A COLLOC RMB     1
                 *
                        END
ERRORS ØØØØØ
```

```
ØØ1   DSPL   ***  LED DISPLAY SUBROUTINE  ***
                            NAM    DSPL     VER 2    12-14-77      CLA
                      *
                      *  FILE NAMES:  &DSPL (S) / DSPL (R)
                      *
                            OPT    REL
                      *
                            TTL    ***  LED DISPLAY SUBROUTINE  ***
                      *
                      **********************************************
                      *
                      *    SUBROUTINE TO DISPLAY DIGITS INPUT VIA THE
                      *    KEYBOARD ON THE SYSTEMS LED'S.  CALLED FROM
                      *     KBIN.
                      *
                      **********************************************
                      *
                      *
ØØØØ CE ØØØØ   A DISPLA LDX    #DIGITØ  ADDR. OF 1ST DIG -1 INTO X
ØØØ3 7F ØØØØ   D        CLR    DSPCNT   CLR DES.PT. COUNTER
ØØØ6 86 Ø1     A        LDAA   #1
ØØØ8 B7 ØØØØ   A        STAA   P8BP     ENABLE 1ST LED
ØØØB 5F                 CLRB
ØØØC 53          DIGLOP COMB
ØØØD F7 ØØØØ   A        STAB   P8AP     ENABLE ALL SEGMENTS
ØØ1Ø F6 ØØØØ   D        LDAB   DSPCNT
ØØ13 C1 Ø6     A        CMPB   #6       6 DIGITS WRITTEN?
ØØ15 26 Ø1 ØØ18         BNE    NOEXIT   NO- DON'T RETURN
ØØ17 39                 RTS             OTHERWISE RETURN
                      *
ØØ18 Ø8          NOEXIT INX
ØØ19 E6 ØØ     A        LDAB   X
ØØ1B F7 ØØØØ   A        STAB   P8AP     PUT DIG ON LED
ØØ1E 5F                 CLRB
ØØ1F 5C          DELAY  INCB
ØØ2Ø 26 FD ØØ1F         BNE    DELAY
ØØ22 48                 ASLA
ØØ23 B7 ØØØØ   A        STAA   P8BP     ENABLE NEXT LED
ØØ26 7C ØØØØ   D        INC    DSPCNT   INC COUNTER
ØØ29 2Ø E1 ØØØC         BRA    DIGLOP   LOOP TO BEGINNING
                      *
                      *
                      *
                        XREF   P8AP,P8BP,DIGITØ
                      *
                      *
                        XDEF   DISPLA
                      *
ØØØØ                    DSCT
                      *
ØØØØ     ØØØ1  A DSPCNT RMB    1
                      *
                        END
ERRORS ØØØØØ
```

# SECTION III:

## Main System Control Routines

```
                        NAM    RESET    VER. 26   9-14-79  CLAVELL
                  *
                  *  FILE NAMES:  &REST (S) / REST (R) /
                  *
                        OPT    REL
                  *
                        TTL    ***  SYSTEM SET UP ROUTINE  ***
                  *
                  ***********************************************
                  *
                  *    THIS ROUTINE INITIALIZES ALL THE PIA'S AND C
                  *    AND THEN SETS THE SYSTEM TO ITS INITIAL COND
                  *
                  ***********************************************
                  *
                  *
                  *
 0000                   ASCT
                  *
                  ***  ASSIGN PIA LOCATIONS  ***
                  *
      7000   A    P1AP  EQU    $7000
      7001   A    P1BP  EQU    $7001     A/D & MULTIPLEXER
      7002   A    P1AC  EQU    $7002
      7003   A    P1BC  EQU    $7003
                  *
      7004   A    P2AP  EQU    $7004     STATUS LED'S
      7005   A    P2BP  EQU    $7005     & D/A
      7006   A    P2AC  EQU    $7006
      7007   A    P2BC  EQU    $7007
                  *
      7008   A    P3AP  EQU    $7008
      7009   A    P3BP  EQU    $7009     STATUS LED DISPLAYS
      700A   A    P3AC  EQU    $700A
      700B   A    P3BC  EQU    $700B
                  *
      700C   A    P4AP  EQU    $700C
      700D   A    P4BP  EQU    $700D     VALVES
      700E   A    P4AC  EQU    $700E
      700F   A    P4BC  EQU    $700F
                  *
      7020   A    P5AP  EQU    $7020
      7021   A    P5BP  EQU    $7021     DEPTH & TEMP
      7022   A    P5AC  EQU    $7022
      7023   A    P5BC  EQU    $7023
                  *
      7024   A    P6AP  EQU    $7024
      7025   A    P6BP  EQU    $7025     LED-CYCLE
      7026   A    P6AC  EQU    $7026     OUTPUTS TO E.T. TIMER
      7027   A    P6BC  EQU    $7027
                  *
      7010   A    P7AP  EQU    $7010
      7011   A    P7BP  EQU    $7011     KEYBOARD INPUT
      7012   A    P7AC  EQU    $7012     & CLOCK INPUT
      7013   A    P7BC  EQU    $7013
                  *
      701C   A    P8AP  EQU    $701C
      701D   A    P8BP  EQU    $701D     6-LED DISPLAY
```

```
002   RESET  ***  SYSTEM SET UP ROUTINE  ***

         701E  A P8AC    EQU    $701E
         701F  A P8BC    EQU    $701F
                 *
 0000                    DSCT
                 *
 0000    0001  A YRH     RMB    1         LOC. FOR YEAR VALUE
 0001    0001  A YRL     RMB    1
 0002    0001  A DAYH    RMB    1         LOC. FOR DAY VALUE
 0003    0001  A DAYL    RMB    1
                 *
                 *   :::  PROGRAM SECTION  :::
                 *
 0000                    PSCT
                 *
                 *   **  RESET & INITIALIZE ALL PIA'S  **
                 *   **  FLAGS AND COUNTERS  **
                 *
 0000 8E 0F90  A START   LDS    #$0F90    INIT. THE STACK
                 *
 0003 0F                 SEI              SET INTERRUPT MASK
                 *
 0004 C6 20    A         LDAB   #32
 0006 86 00    A         LDAA   #$00      INIT. ALL PIA'S
 0008 CE 7000  A         LDX    #P1AP
 000B A7 00    A LOOP1   STAA   X         FILL WITH 0'S
 000D 08                 INX
 000E 5A                 DECB
 000F 26 FA 000B         BNE    LOOP1
                 *
 0011 86 F0    A         LDAA   #$F0      PIA 1 (PA0-7,PB0-3:INPUTS)
 0013 B7 7001  A         STAA   P1BP      (PB4-7:OUTPUTS)
                 *
 0016 86 FF    A         LDAA   #$FF
 0018 B7 7004  A         STAA   P2AP      ALL OUTPUTS
 001B B7 7005  A         STAA   P2BP
                 *
 001E B7 7008  A         STAA   P3AP      ALL OUTPUTS
 0021 B7 7009  A         STAA   P3BP
                 *
 0024 B7 700C  A         STAA   P4AP      ALL OUTPUTS
 0027 B7 700D  A         STAA   P4BP
                 *
 002A B7 7021  A         STAA   P5BP      PA0-7:INPUTS
                 *                          PB0-7:OUTPUTS
                 *
 002D B7 7024  A         STAA   P6AP      ALL OUTPUTS
 0030 B7 7025  A         STAA   P6BP
                 *
                 *              PIA 7:  ALL INPUTS
                 *
 0033 B7 701C  A         STAA   P8AP
 0036 B7 701D  A         STAA   P8BP      ALL OUTPUTS
                 *
                 *   ** SET UP CONTROL REGISTERS  **
                 *
 0039 C6 34    A         LDAB   #$34
 003B 86 06    A         LDAA   #$06
                 *
```

```
003  RESET  ***  SYSTEM SET UP ROUTINE  ***

003D F7 7002  A        STAB   P1AC
0040 B7 7003  A        STAA   P1BC
                *
                *
0043 B7 7022  A        STAA   P5AC
                *
0046 B7 7012  A        STAA   P7AC
0049 F7 7013  A        STAB   P7BC
                *
004C B7 701E  A        STAA   P8AC
004F B7 701F  A        STAA   P8BC
                *
0052 86 0D    A        LDAA   #$0D     CA1&CA2,CB1&CB2 SET UP
0054 B7 700A  A        STAA   P3AC     FOR NEG. EDGE & UNMASKED
0057 B7 700B  A        STAA   P3BC     THESE ARE FOR OVER-RIDE
                *                   FLAG INTERRUPTS
005A B7 700E  A        STAA   P4AC
005D B7 700F  A        STAA   P4BC
                *
0060 B6 7008  A        LDAA   P3AP     DUMMY READS
0063 B6 7009  A        LDAA   P3BP     TO CLR INTERRUPT
0066 B6 700C  A        LDAA   P4AP     FLAGS
0069 B6 700D  A        LDAA   P4BP
                *
006C 86 06    A        LDAA   #%00000110 CA1 &CA2 MASKED
006E B7 7006  A        STAA   P2AC
                *
0071 86 07    A        LDAA   #$07     CB1 UNMASKED (OVR RESET)
0073 B7 7007  A        STAA   P2BC     CB2 MASKED
                *
0076 B6 7004  A        LDAA   P2AP     DUMMY READS
0079 B6 7005  A        LDAA   P2BP
                *
007C C6 3E    A        LDAB   #$3E
007E F7 7026  A        STAB   P6AC     SET CONTROL REG
0081 C6 36    A        LDAB   #$36     FOR E.T. CONTROL
0083 F7 7027  A        STAB   P6BC
                *
0086 86 1C    A        LDAA   #$1C     CB1 MASKED (GO INTERRUPT)
0088 B7 7023  A        STAA   P5BC     CB2 UNMASKED (HALT INTERRU
                *
                *
008B 86 60    A        LDAA   #%01100000 LIFT PEN & SCAN OFF
008D B7 7005  A        STAA   P2BP
                *
0090 86 25    A        LDAA   #%00100101 SET VALVES: V1A,V2A,V3B
0092 B7 700C  A        STAA   P4AP
                *
0095 86 03    A        LDAA   #3       1.5 SEC DELAY
0097 BD 0000  A        JSR    GENTIM
                *
009A 7F 700C  A        CLR    P4AP     RESET TIMER
009D 86 A8    A        LDAA   #%10101000 SET VALVES: V7B,V8B,V6B
009F B7 700D  A        STAA   P4BP
                *
00A2 86 03    A        LDAA   #3
00A4 BD 0000  A        JSR    GENTIM
                *
```

```
004   RESET  ***  SYSTEM SET UP ROUTINE  ***

 00A7 7F 700D  A           CLR    P4BP      RESET TIMER
                 *
 00AA 86 C0    A           LDAA   #$C0      SET CYCLE LED
 00AC B7 7024  A           STAA   P6AP      TO 0
                 *
 00AF 7F 7025  A           CLR    P6BP      CLR .E.T.
 00B2 7F 701D  A           CLR    P8BP      OFF LED SEG
                 *
                 *         ::: CLEAR ALL FLAGS  :::
                 *
 00B5 7F 0000  A           CLR    POLFLG
 00B8 7F 0000  A           CLR    NUMBYH
 00BB 7F 0000  A           CLR    NUMBYL
 00BE 7F 0039  D           CLR    PRXFLG
 00C1 7F 003A  D           CLR    ACDFLG
 00C4 7F 0008  D           CLR    RCLFLG
 00C7 7F 000D  D           CLR    VAFLG
 00CA 7F 000E  D           CLR    VBFLG
 00CD 7F 000F  D           CLR    RESTFG
 00D0 7F 0010  D           CLR    TIMFLG
 00D3 7F 0012  D           CLR    CNTR2
 00D6 7F 0000  A           CLR    ASCBUF
 00D9 7F 0000  A           CLR    OVR
 00DC 7F 0000  A           CLR    FRSTQF    DEC./INTEGER FLAG
 00DF 7F 0000  A           CLR    SAMPLE    USED IN COMPT
 00E2 7F 0000  A           CLR    ERFLAG     "    "   "
                 *
 00E5 0E                   CLI
                 *
 00E6 86 0F    A           LDAA   #$0F
 00E8 B7 7025  A           STAA   P6BP      SET E.T. TO 1 SEC.
 00EB BD 0000  A           JSR    CT
 00EE C6 00    A           LDAB   #0
 00F0 F7 7025  A           STAB   P6BP
 00F3 BD 0000  A           JSR    CT        DUMMY DIGIT
 00F6 86 01    A           LDAA   #1        LSD = 1
 00F8 B7 7025  A           STAA   P6BP
 00FB BD 0000  A           JSR    CT
 00FE F7 7025  A           STAB   P6BP      SET OTHER DIGITS
 0101 BD 0000  A           JSR    CT        TO 0
 0104 BD 0000  A           JSR    CT
 0107 BD 0000  A           JSR    CT
 010A BD 0000  A           JSR    CT
 010D BD 0000  A           JSR    CT
                 *
 0110 C6 28    A           LDAB   #40       BOOT ERROR MESG
 0112 CE 0000  A           LDX    #MESER    INTO RAM
 0115 FF 003B  D           STX    TMES
 0118 CE 0013  D           LDX    #MESERR
 011B FF 003D  D           STX    T1MES
                 *
 011E FE 003B  D           LDX    TMES
 0121 A6 00    A BOOT      LDAA   0,X
 0123 FE 003D  D           LDX    T1MES
 0126 A7 00    A           STAA   0,X
 0128 08                   INX
 0129 FF 003D  D           STX    T1MES
 012C FE 003B  D           LDX    TMES
```

```
005   RESET  ***  SYSTEM SET UP ROUTINE  ***

 012F 08                    INX
 0130 FF 003B  D            STX    TMES
 0133 5A                    DECB
 0134 26 EB 0121            BNE    BOOT
                  *
 0136 7D 0000  A            TST    NUMBYL
 0139 26 16 0151            BNE    TABLE
                  *
 013B CE 0000  A            LDX    #TTBL1    BOOT IN 7-SEG TABLE
 013E FF 003B  D            STX    TMES
 0141 CE 0010  B            LDX    #TBL1
 0144 FF 003D  D            STX    T1MES
 0147 C6 0A    A            LDAB   #10
 0149 7C 0000  A            INC    NUMBYL
 014C FE 003B  D            LDX    TMES
 014F 20 D0 0121            BRA    BOOT
                  *
 0151 B6 0000  A TABLE      LDAA   NUMBYL
 0154 81 02    A            CMPA   #2
 0156 27 17 016F            BEQ    OUT
 0158 7C 0000  A            INC    NUMBYL
 015B C6 10    A            LDAB   #16       BOOT IN OTHER TABLE
 015D CE 0000  A            LDX    #TTBL
 0160 FF 003B  D            STX    TMES
 0163 CE 0000  B            LDX    #TBL
 0166 FF 003D  D            STX    T1MES
 0169 FE 003B  D            LDX    TMES
 016C 7E 0121  P            JMP    BOOT
                  *
 016F 7F 0000  A OUT        CLR    NUMBYL
                  *
 0172 CE 0000  A            LDX    #MES1     1 ST MESSAGE: YEAR?
 0175 BD 0000  A            JSR    PRINT
                  *
 0178 86 04    A            LDAA   #4        TAKE IN DATA
 017A BD 0000  A            JSR    KBIN      CONV, TO BIN
 017D B6 0000  A            LDAA   BINLO     & STORE
 0180 F6 0000  A            LDAB   BINHI
 0183 B7 0001  D            STAA   YRL
 0186 F7 0000  D            STAB   YRH
                  *
 0189 CE 0000  A            LDX    #MES2     ASK FOR JULIAN DATE
 018C BD 0000  A            JSR    PRINT
                  *
 018F 86 04    A            LDAA   #4
 0191 BD 0000  A            JSR    KBIN      INPUT
 0194 B6 0000  A            LDAA   BINLO
 0197 F6 0000  A            LDAB   BINHI
 019A B7 0004  D            STAA   DAYL0
 019D F7 0005  D            STAB   DAYH0
                  *
 01A0 BD 0000  A            JSR    SUB3      SPACE
 01A3 CE 0000  A            LDX    #MES3     ASK OP. TO SET CLOCK
 01A6 C6 02    A            LDAB   #2
 01A8 F7 0000  A            STAB   BLOCK
 01AB BD 0000  A            JSR    PRINT1
 01AE BD 0000  A            JSR    SUB3      SPACE
                  *
```

```
ØØ6   RESET  ***  SYSTEM SET UP ROUTINE  ***

 Ø1B1 BD ØØØØ  A         JSR    TIME     DISPLAY TIME ON LED'S
                *                      UNTIL GO PUSHED
                *
                *
                *
 Ø1B4 7E ØØØØ  A         JMP    INIT
                **
                *
                *      *** CHECK IF "GO" BUTTON HAS BEEN PUSHED  *
                *
 Ø1B7 B6 7Ø21  A RETSET LDAA   P5BP     DUMMY READ
 Ø1BA B6 7Ø23  A GOLOOP LDAA   P5BC     POLL GO BUTTON
 Ø1BD 2A FB Ø1BA         BPL    GOLOOP   NOT PUSHED-LOOP
                *                         OTHERWISE JMP TO
 Ø1BF 7F ØØØØ  A         CLR    INTFLG   MAIN CONTROL PROGM
 Ø1C2 7E ØØØØ  A         JMP    PROGM
                *
                *
                *
                         XREF   TIME,PRINT,MES1,MES2,MES3,PROGM
                         XREF   BINLO,BINHI,ASCBUF,GENTIM,PRINT1
                         XREF   NUMBYH,NUMBYL,POLFLG,OVR,CT,FRSTQF
                         XREF   SAMPLE,BLOCK,KBIN,ERFLAG
                         XREF   MESER,INTFLG,INIT,SUB3,TTBL,TTBL1
                *
                *
                *
                         XDEF   P1AP,P1BP,P1AC,P1BC,P2AP,P2BP,P2AC
                         XDEF   P3AP,P3BP,P3AC,P3BC,P4AP,P4BP,P4AC
                         XDEF   P5AP,P5BP,P5AC,P5BC,P6AP,P6BP,P6AC
                         XDEF   P7AP,P7BP,P7AC,P7BC,P8AP,P8BP,P8AC
                         XDEF   YRH,YRL,DAYH,DAYL,EMPFLG
                         XDEF   TMFLG,PRXFLG,ACDFLG,RCLFLG,SCNFLG
                         XDEF   CNTR1,CNTR2,START,RETSET,MESERR
                         XDEF   P2BC,P4BC,P6BC,P8BC,TIMFLG
                         XDEF   TMES,T1MES,TBL,TBL1,DAYLØ,DAYHØ
                *
                *
 ØØØ4                    DSCT
                *
 ØØØ4      ØØØ1  A DAYLØ  RMB    1
 ØØØ5      ØØØ1  A DAYHØ  RMB    1
 ØØØ6      ØØØ1  A EMPFLG RMB    1
 ØØØ7      ØØØ1  A TMFLG  RMB    1
 ØØØ8      ØØØ1  A RCLFLG RMB    1
 ØØØ9      ØØØ1  A PRXFLG RMB    1
 ØØØA      ØØØ1  A ACDFLG RMB    1
 ØØØB      ØØØ1  A SCNFLG RMB    1
 ØØØC      ØØØ1  A STDFLG RMB    1
 ØØØD      ØØØ1  A VAFLG  RMB    1
 ØØØE      ØØØ1  A VBFLG  RMB    1
 ØØØF      ØØØ1  A RESTFG RMB    1
 ØØ1Ø      ØØØ1  A TIMFLG RMB    1
 ØØ11      ØØØ1  A CNTR1  RMB    1
 ØØ12      ØØØ1  A CNTR2  RMB    1
 ØØ13      ØØ28  A MESERR RMB    4Ø
 ØØ3B      ØØØ2  A TMES   RMB    2
 ØØ3D      ØØØ2  A T1MES  RMB    2
```

```
007   RESET   ***  SYSTEM SET UP ROUTINE  ***
                              *
 0000                                BSCT
                              *
 0000      0010   A TBL      RMB      16
 0010      000A   A TBL1     RMB      10
                              *
                                     END
ERRORS 00000
```

```
                                    21      3-7-80
                         NAM    INIT     VER. ~~20~~  ~~9-14-79~~  CLAVEL
                    *
                    *  FILE NAMES:   SINIT (S) / INIT (R) /
                    *
                         OPT    REL
                    *
                         TTL    ***  INITIALIZATION SUBROUTINE  ***
                    *
                    ***********************************************
                    *
                    *   THIS SUBROUTINE INITIALIZES THE INSTRUMENT P
                    *   AND INPUTS PARAMETER DATA.
                    *
                    ***********************************************
                    *
 0000                    BSCT
                    *
 0000      0002   A STCN1H RMB    2        ZN STD CONC. BUFFER
 0002      0002   A STCN2H RMB    2        CD
 0004      0002   A STCN3H RMB    2        PB
 0006      0002   A STCN4H RMB    2        CU
                    *
 0000                    PSCT
                    *
                    *
 0000 7F 0011   D INIT   CLR    ZINK
 0003 7F 0012   D        CLR    VPRECY
 0006 7F 0000   A        CLR    F2
 0009 7F 0003   D        CLR    INTFLG
 000C 7C 0003   D        INC    INTFLG
                    *
 000F CE 00DA   P        LDX    #A0      INITIALIZE JUMP TABLE
 0012 FF 0008   B        STX    J0       IN BASE SECTION
 0015 CE 00FC   P        LDX    #A1
 0018 FF 000A   B        STX    J0+2
 001B CE 011E   P        LDX    #A2
 001E FF 000C   B        STX    J0+4
 0021 CE 013A   P        LDX    #A3
 0024 FF 000E   B        STX    J0+6
 0027 CE 0156   P        LDX    #A4
 002A FF 0010   B        STX    J0+8
 002D CE 0179   P        LDX    #A5
 0030 FF 0012   B        STX    J0+10
 0033 CE 01A4   P        LDX    #A6
 0036 FF 0014   B        STX    J0+12
 0039 CE 01D4   P        LDX    #A7
 003C FF 0016   B        STX    J0+14
 003F CE 01F5   P        LDX    #A8
 0042 FF 0018   B        STX    J0+16
 0045 CE 0221   P        LDX    #A9
 0048 FF 001A   B        STX    J0+18
 004B CE 0243   P        LDX    #A10
 004E FF 001C   B        STX    J0+20
 0051 CE 025F   P        LDX    #A11
 0054 FF 001E   B        STX    J0+22
 0057 CE 027B   P        LDX    #A12
 005A FF 0020   B        STX    J0+24
 005D CE 02A6   P        LDX    #A13
```

```
ØØ6Ø FF ØØ22  B          STX    JØ+26
ØØ63 CE Ø2CC  P          LDX    #A14
ØØ66 FF ØØ24  B          STX    JØ+28
ØØ69 CE Ø2F7  P          LDX    #A15
ØØ6C FF ØØ26  B          STX    JØ+3Ø
ØØ6F CE Ø31D  P          LDX    #A16
ØØ72 FF ØØ28  B          STX    JØ+32
ØØ75 CE Ø343  P          LDX    #A17
ØØ78 FF ØØ2A  B          STX    JØ+34
ØØ7B CE Ø36D  P          LDX    #A18
ØØ7E FF ØØ2C  B          STX    JØ+36
ØØ81 CE Ø397  P          LDX    #A19
ØØ84 FF ØØ2E  B          STX    JØ+38
ØØ87 CE Ø3C1  P          LDX    #A2Ø
ØØ8A FF ØØ3Ø  B          STX    JØ+4Ø
ØØ8D CE Ø3FØ  P          LDX    #A21
ØØ9Ø FF ØØ32  B          STX    JØ+42
ØØ93 CE Ø41A  P          LDX    #A22
ØØ96 FF ØØ34  B          STX    JØ+44
ØØ99 CE Ø43F  P          LDX    #A23
ØØ9C FF ØØ36  B          STX    JØ+46
ØØ9F CE Ø469  P          LDX    #A24
ØØA2 FF ØØ38  B          STX    JØ+48
ØØA5 CE ØØØØ  A          LDX    #FRSTQ
ØØA8 FF ØØ3A  B          STX    JØ+5Ø
ØØAB CE Ø48E  P          LDX    #RENTR
ØØAE FF ØØ3C  B          STX    JØ+52
ØØB1 CE Ø4AB  P          LDX    #A27
ØØB4 FF ØØ3E  B          STX    JØ+54
ØØB7 CE Ø4CC  P          LDX    #A28
ØØBA FF ØØ4Ø  B          STX    JØ+56
                *
                *
ØØBD 7E ØØØØ  A          JMP    FRSTQ     GET IP & FP
                *
ØØCØ 7F ØØØØ  A RETINI   CLR    P4AP      RESET TIMER
                *
                *
                *
ØØC3 CE ØØØØ  A          LDX    #MES14
ØØC6 C6 Ø4    A          LDAB   #4
ØØC8 F7 ØØØØ  A          STAB   BLOCK     SYSTEM INIT. MESG.
ØØCB BD ØØØØ  A          JSR    SUB3
ØØCE BD ØØØØ  A          JSR    SUB3
ØØD1 BD ØØØØ  A          JSR    PRINT1
                *
ØØD4 BD ØØØØ  A          JSR    SUB3
ØØD7 BD ØØØØ  A          JSR    SUB3
                *
ØØDA CE ØØØØ  A AØ       LDX    #MES16    BEGIN ASKING
ØØDD BD ØØØØ  A          JSR    PRINT     VP QUESTIONS
ØØEØ 86 Ø3    A          LDAA   #3        &
ØØE2 BD ØØØØ  A          JSR    KBIN      GET INPUT VALUES
ØØE5 B6 ØØØØ  A          LDAA   BCDHI
ØØE8 F6 ØØØØ  A          LDAB   BCDLO
ØØEB B7 ØØØ6  D          STAA   VPHGH
ØØEE F7 ØØØ7  D          STAB   VPHGL     HG PLATTING TIME
ØØF1 BD ØØØØ  A          JSR    SUB3
```

```
003   INIT   ***  INITIALIZATION SUBROUTINE  ***

                     *
 00F4 7D 0000  A            TST     F2         CALLED FROM OFIX?
 00F7 27 03 00FC            BEQ     A1         NO: CONT.
 00F9 7E 0000  A            JMP     RTNPT      YES: RETURN TO OFIX
                     *
 00FC CE 0000  A A1         LDX     #MES17
 00FF BD 0000  A            JSR     PRINT
 0102 86 03    A            LDAA    #~~3~~ 4 low
 0104 BD 0000  A            JSR     KBIN       SAMPLE TIME
 0107 B6 0000  A            LDAA    BCDHI
 010A F6 0000  A            LDAB    BCDLO
 010D B7 0008  D            STAA    VPSAMH
 0110 F7 0009  D            STAB    VPSAML
 0113 BD 0000  A            JSR     SUB3
                     *
 0116 7D 0000  A            TST     F2
 0119 27 03 011E            BEQ     A2
 011B 7E 0000  A            JMP     RTNPT
                     *
 011E CE 0000  A A2         LDX     #MES18
 0121 BD 0000  A            JSR     PRINT
 0124 86 01    A            LDAA    #1         VALVE DELAY TIME
 0126 BD 0000  A            JSR     KBIN
 0129 B6 0000  A            LDAA    BCDLO
 012C B7 000A  D            STAA    VPVS1
 012F BD 0000  A            JSR     SUB3
                     *
 0132 7D 0000  A            TST     F2
 0135 27 03 013A            BEQ     A3
 0137 7E 0000  A            JMP     RTNPT
                     *
 013A CE 0000  A A3         LDX     #MES19
 013D BD 0000  A            JSR     PRINT
 0140 86 01    A            LDAA    #1         VALVE DELAY2 TIME
 0142 BD 0000  A            JSR     KBIN
 0145 B6 0000  A            LDAA    BCDLO
 0148 B7 000B  D            STAA    VPVS2
 014B BD 0000  A            JSR     SUB3
                     *
 014E 7D 0000  A            TST     F2
 0151 27 03 0156            BEQ     A4
 0153 7E 0000  A            JMP     RTNPT
                     *
 0156 CE 0000  A A4         LDX     #MES20
 0159 BD 0000  A            JSR     PRINT
 015C 86 03    A            LDAA    #3         SCAN TIME
 015E BD 0000  A            JSR     KBIN
 0161 B6 0000  A            LDAA    BCDHI
 0164 F6 0000  A            LDAB    BCDLO
 0167 B7 0004  D            STAA    SCANTH
 016A F7 0005  D            STAB    SCANTL
 016D BD 0000  A            JSR     SUB3
                     *
 0170 7D 0000  A            TST     F2
 0173 27 03 0178            BEQ     A5
 0175 7E 0000  A            JMP     RTNPT
                     *
 0178 CE 0000  A A5         LDX     #MES21
```

```
017B BD 0000  A        JSR    PRINT
017E 86 03    A        LDAA   #3        FLUSHING TIME
0180 BD 0000  A        JSR    KBIN
0183 B6 0000  A        LDAA   BINLO
                *
0186 81 78    A        CMPA   #120
0188 25 0B 0195        BCS    STA3
                *
018A CE 0000  A        LDX    #MES65
018D BD 0000  A        JSR    PRINT1
0190 BD 0000  A        JSR    SUB3
0193 20 E3 0178        BRA    A5
                *
0195 48         STA3   ASLA             *2
0196 B7 000C  D        STAA   VPFLSL
0199 BD 0000  A        JSR    SUB3
                *
019C 7D 0000  A        TST    F2
019F 27 03 01A4        BEQ    A6
01A1 7E 0000  A        JMP    RTNPT
                *
01A4 CE 0000  A A6     LDX    #MES22
01A7 BD 0000  A        JSR    PRINT
01AA 86 03    A        LDAA   #3
01AC BD 0000  A        JSR    KBIN      STD ADD TIME
01AF B6 0000  A        LDAA   BINLO     IN BIN (0-255 SEC MAX)
                *
01B2 81 01    A        CMPA   #1        # MUST BE > 1 & < 120
01B4 23 04 01BA        BLS    ER10
                *
01B6 81 78    A        CMPA   #120
01B8 25 0B 01C5        BCS    STA1
                *
01BA CE 0000  A ER10   LDX    #MES65    ERROR MESG
01BD BD 0000  A        JSR    PRINT1
01C0 BD 0000  A        JSR    SUB3
01C3 20 DF 01A4        BRA    A6
                *
01C5 48         STA1   ASLA             "A" * 2
01C6 B7 000D  D        STAA   VPSTD
01C9 BD 0000  A        JSR    SUB3
                *
01CC 7D 0000  A        TST    F2
01CF 27 03 01D4        BEQ    A7
01D1 7E 0000  A        JMP    RTNPT
                *
01D4 CE 0000  A A7     LDX    #MES40
01D7 BD 0000  A        JSR    PRINT     MAKE ACID ADDITION?
01DA 86 01    A        LDAA   #1
01DC BD 0000  A        JSR    KBIN
01DF B6 0000  A        LDAA   BINLO
01E2 B7 0000  A        STAA   ACDFLG
01E5 BD 0000  A        JSR    SUB3
                *
01E8 7D 0000  A        TST    F2
01EB 27 03 01F0        BEQ    S1
01ED 7E 0000  A        JMP    RTNPT
                *
```

```
3Ø5   INIT   ***  INITIALIZATION SUBROUTINE  ***

 Ø1FØ 7D ØØØØ  A S1     TST    ACDFLG    MAKE ACID ADD.?
 Ø1F3 27 2C Ø221        BEQ    A9        NO: DON'T ADD
                 *
 Ø1F5 CE ØØØØ  A A8     LDX    #MES23
 Ø1F8 BD ØØØØ  A        JSR    PRINT
 Ø1FB 86 Ø3    A        LDAA   #3        ACID ADD TIME
 Ø1FD BD ØØØØ  A        JSR    KBIN      INPUT BIN
 Ø2ØØ B6 ØØØØ  A        LDAA   BINLO     (Ø-255 SEC)
                 *
 Ø2Ø3 81 78    A        CMPA   #12Ø
 Ø2Ø5 25 ØB Ø212        BCS    STA2
                 *
 Ø2Ø7 CE ØØØØ  A        LDX    #MES65
 Ø2ØA BD ØØØØ  A        JSR    PRINT1
 Ø2ØD BD ØØØØ  A        JSR    SUB3
 Ø21Ø 2Ø E3 Ø1F5        BRA    A8
                 *
 Ø212 48         STA2   ASLA             *2
 Ø213 B7 ØØØE  D        STAA   VPACID
 Ø216 BD ØØØØ  A        JSR    SUB3
                 *
 Ø219 7D ØØØØ  A        TST    F2
 Ø21C 27 Ø3 Ø221        BEQ    A9
 Ø21E 7E ØØØØ  A        JMP    RTNPT
                 *
 Ø221 CE ØØØØ  A A9     LDX    #MES24
 Ø224 BD ØØØØ  A        JSR    PRINT
 Ø227 86 Ø3    A        LDAA   #3        CO2 PURGE TIME
 Ø229 BD ØØØØ  A        JSR    KBIN
 Ø22C B6 ØØØØ  A        LDAA   BCDHI
 Ø22F F6 ØØØØ  A        LDAB   BCDLO
 Ø232 B7 ØØØF  D        STAA   VPPRGH
 Ø235 F7 ØØ1Ø  D        STAB   VPPRGL
 Ø238 BD ØØØØ  A        JSR    SUB3
                 *
 Ø23B 7D ØØØØ  A        TST    F2
 Ø23E 27 Ø3 Ø243        BEQ    A1Ø
 Ø24Ø 7E ØØØØ  A        JMP    RTNPT
                 *
 Ø243 CE ØØØØ  A A1Ø    LDX    #MES25
 Ø246 BD ØØØØ  A        JSR    PRINT
 Ø249 86 Ø1    A        LDAA   #1
 Ø24B BD ØØØØ  A        JSR    KBIN      ZINK ANALYSIS?
 Ø24E B6 ØØØØ  A        LDAA   BINLO     YES=1,NO=Ø
 Ø251 B7 ØØ11  D        STAA   ZINK
 Ø254 BD ØØØØ  A        JSR    SUB3
                 *
 Ø257 7D ØØØØ  A        TST    F2
 Ø25A 27 Ø3 Ø25F        BEQ    A11
 Ø25C 7E ØØØØ  A        JMP    RTNPT
                 *
 Ø25F CE ØØØØ  A A11    LDX    #MES26
 Ø262 BD ØØØØ  A        JSR    PRINT
 Ø265 86 Ø1    A        LDAA   #1        RECIRCULATE MODE?
 Ø267 BD ØØØØ  A        JSR    KBIN      YES=1,NO=Ø
 Ø26A B6 ØØØØ  A        LDAA   BINLO
 Ø26D B7 ØØØØ  A        STAA   RCLFLG
 Ø27Ø BD ØØØØ  A        JSR    SUB3
```

```
                          *
 0273 7D 0000  A                 TST     F2
 0276 27 03 027B                 BEQ     A12
 0278 7E 0000  A                 JMP     RTNPT
                          *
 027B CE 0000  A A12             LDX     #MES27
 027E BD 0000  A                 JSR     PRINT
 0281 86 01    A                 LDAA    #1          STD ADD CYCLE #
 0283 BD 0000  A                 JSR     KBIN
 0286 B6 0000  A                 LDAA    BINLO
                          *
 0289 81 01    A                 CMPA    #1          MUST BE > 1
 028B 2E 0B 0298                 BGT     STA
                          *
 028D CE 0000  A                 LDX     #MES65      ERROR MESSG
 0290 BD 0000  A                 JSR     PRINT1
 0293 BD 0000  A                 JSR     SUB3
 0296 20 E3 027B                 BRA     A12
                          *
 0298 B7 0012  D STA             STAA    VPRECY
 029B BD 0000  A                 JSR     SUB3
                          *
 029E 7D 0000  A                 TST     F2
 02A1 27 03 02A6                 BEQ     A13
 02A3 7E 0000  A                 JMP     RTNPT
                          *
 02A6 7C 0000  A A13             INC     FRSTOF      DEC. INPUT
                          *
 02A9 CE 0000  A                 LDX     #MES7
 02AC BD 0000  A                 JSR     PRINT
 02AF 86 03    A                 LDAA    #3          CD STD CONC.
 02B1 BD 0000  A                 JSR     KBIN
 02B4 B6 0000  A                 LDAA    BINHI
 02B7 F6 0000  A                 LDAB    BINLO
 02BA 97 02    B                 STAA    STCN2H
 02BC D7 03    B                 STAB    STCN2H+1
 02BE BD 0000  A                 JSR     SUB3
                          *
 02C1 7D 0000  A                 TST     F2
 02C4 27 06 02CC                 BEQ     A14
 02C6 7F 0000  A                 CLR     FRSTOF
 02C9 7E 0000  A                 JMP     RTNPT
                          *
 02CC 7C 0000  A A14             INC     FRSTOF
 02CF CE 0000  A                 LDX     #MES9
 02D2 BD 0000  A                 JSR     PRINT
 02D5 86 03    A                 LDAA    #3          PB STD CONC.
 02D7 BD 0000  A                 JSR     KBIN
 02DA B6 0000  A                 LDAA    BINHI
 02DD F6 0000  A                 LDAB    BINLO
 02E0 97 04    B                 STAA    STCN3H
 02E2 D7 05    B                 STAB    STCN3H+1
 02E4 BD 0000  A                 JSR     SUB3
                          *
 02E7 7D 0000  A                 TST     F2
 02EA 27 0B 02F7                 BEQ     A15
 02EC 7F 0000  A                 CLR     FRSTOF
 02EF 7E 0000  A                 JMP     RTNPT
```

```
307   INIT    ***  INITIALIZATION SUBROUTINE  ***
                        *
 02F2 7D 0011  D            TST     ZINK
 02F5 26 26 031D            BNE     A16
                        *
 02F7 7C 0000  A  A15       INC     FRSTOF
 02FA CE 0000  A            LDX     #MES10
 02FD BD 0000  A            JSR     PRINT
 0300 86 03    A            LDAA    #3         CU STD
 0302 BD 0000  A            JSR     KBIN
 0305 B6 0000  A            LDAA    BINHI
 0308 F6 0000  A            LDAB    BINLO
 030B 97 06    B            STAA    STCN4H
 030D D7 07    B            STAB    STCN4H+1
 030F BD 0000  A            JSR     SUB3
                        *
 0312 7D 0000  A            TST     F2
 0315 27 2C 0343            BEQ     A17
 0317 7F 0000  A            CLR     FRSTOF
 031A 7E 0000  A            JMP     RTNPT
                        *
 031D 7C 0000  A  A16       INC     FRSTOF
 0320 CE 0000  A            LDX     #MES11
 0323 BD 0000  A            JSR     PRINT
 0326 86 03    A            LDAA    #3         ZN STD
 0328 BD 0000  A            JSR     KBIN
 032B B6 0000  A            LDAA    BINHI
 032E F6 0000  A            LDAB    BINLO
 0331 97 00    B            STAA    STCN1H
 0333 D7 01    B            STAB    STCN1H+1
 0335 BD 0000  A            JSR     SUB3
                        *
 0338 7D 0000  A            TST     F2
 033B 27 06 0343            BEQ     A17
 033D 7F 0000  A            CLR     FRSTOF
 0340 7E 0000  A            JMP     RTNPT
                        *
                        *
 0343 7C 0000  A  A17       INC     FRSTOF
 0346 CE 0000  A            LDX     #MES70     CD LOWER LIMIT
 0349 BD 0000  A            JSR     PRINT
 034C 86 01    A            LDAA    #1
 034E BD 0000  A            JSR     KBIN
 0351 B6 0000  A            LDAA    BINHI
 0354 84 7F    A            ANDA    #$7F       MASK SIGN BIT
 0356 F6 0000  A            LDAB    BINLO
 0359 B7 0002  A            STAA    CD+2
 035C F7 0003  A            STAB    CD+3
 035F BD 0000  A            JSR     SUB3
                        *
 0362 7D 0000  A            TST     F2
 0365 27 06 036D            BEQ     A18
 0367 7F 0000  A            CLR     FRSTOF
 036A 7E 0000  A            JMP     RNG        RECALCULATE RANGES
                        *
 036D 7C 0000  A  A18       INC     FRSTOF
 0370 CE 0000  A            LDX     #MES71     CD HIGH LIMIT
 0373 BD 0000  A            JSR     PRINT
 0376 86 01    A            LDAA    #1
```

```
0378 BD 0000  A         JSR    KBIN
037B B6 0000  A         LDAA   BINHI
037E 84 7F    A         ANDA   #$7F
0380 F6 0000  A         LDAB   BINLO
0383 B7 0000  A         STAA   CD
0386 F7 0001  A         STAB   CD+1
0389 BD 0000  A         JSR    SUB3
                *
038C 7D 0000  A         TST    F2
038F 27 06 0397         BEQ    A19
0391 7F 0000  A         CLR    FRSTQF
0394 7E 0000  A         JMP    RNG
                *
0397 7C 0000  A A19     INC    FRSTQF
039A CE 0000  A         LDX    #MES72    PB LOWER LIMIT
039D BD 0000  A         JSR    PRINT
03A0 86 01    A         LDAA   #1
03A2 BD 0000  A         JSR    KBIN
03A5 B6 0000  A         LDAA   BINHI
03A8 84 7F    A         ANDA   #$7F
03AA F6 0000  A         LDAB   BINLO
03AD B7 0002  A         STAA   PB+2
03B0 F7 0003  A         STAB   PB+3
03B3 BD 0000  A         JSR    SUB3
                *
03B6 7D 0000  A         TST    F2
03B9 27 06 03C1         BEQ    A20
03BB 7F 0000  A         CLR    FRSTQF
03BE 7E 0000  A         JMP    RNG
                *
03C1 7C 0000  A A20     INC    FRSTQF
03C4 CE 0000  A         LDX    #MES73    PB UPPER LIMIT
03C7 BD 0000  A         JSR    PRINT
03CA 86 01    A         LDAA   #1
03CC BD 0000  A         JSR    KBIN
03CF B6 0000  A         LDAA   BINHI
03D2 84 7F    A         ANDA   #$7F
03D4 F6 0000  A         LDAB   BINLO
03D7 B7 0000  A         STAA   PB
03DA F7 0001  A         STAB   PB+1
03DD BD 0000  A         JSR    SUB3
                *
03E0 7D 0000  A         TST    F2
03E3 27 0B 03F0         BEQ    A21
03E5 7F 0000  A         CLR    FRSTQF
03E8 7E 0000  A         JMP    RNG
                *
03EB 7D 0011  D         TST    ZINK      ZINC ANALYSIS?
03EE 26 4F 043F         BNE    A23       YES: GO TO A23
                *
03F0 7C 0000  A A21     INC    FRSTQF
03F3 CE 0000  A         LDX    #MES74    CU LOWER LIMIT
03F6 BD 0000  A         JSR    PRINT
03F9 86 01    A         LDAA   #1
03FB BD 0000  A         JSR    KBIN
03FE B6 0000  A         LDAA   BINHI
0401 84 7F    A         ANDA   #$7F
0403 F6 0000  A         LDAB   BINLO
```

```
0406 B7 0002  A         STAA   CU+2
0409 F7 0003  A         STAB   CU+3
040C BD 0000  A         JSR    SUB3
               *
040F 7D 0000  A         TST    F2
0412 27 06 041A         BEQ    A22
0414 7F 0000  A         CLR    FRSTQF
0417 7E 0000  A         JMP    RNG
               *
041A 7C 0000  A A22     INC    FRSTQF
041D CE 0000  A         LDX    #MES75   CU UPPER LIMIT
0420 BD 0000  A         JSR    PRINT
0423 86 01    A         LDAA   #1
0425 BD 0000  A         JSR    KBIN
0428 B6 0000  A         LDAA   BINHI
042B 84 7F    A         ANDA   #$7F
042D F6 0000  A         LDAB   BINLO
0430 B7 0000  A         STAA   CU
0433 F7 0001  A         STAB   CU+1
0436 BD 0000  A         JSR    SUB3
               *
0439 7F 0000  A         CLR    FRSTOF
043C 7E 0000  A         JMP    RNG
               *
043F 7C 0000  A A23     INC    FRSTOF
0442 CE 0000  A         LDX    #MES76   ZN LOWER LIMIT
0445 BD 0000  A         JSR    PRINT
0448 86 01    A         LDAA   #1
044A BD 0000  A         JSR    KBIN
044D B6 0000  A         LDAA   BINHI
0450 84 7F    A         ANDA   #$7F
0452 F6 0000  A         LDAB   BINLO
0455 B7 0002  A         STAA   ZN+2
0458 F7 0003  A         STAB   ZN+3
045B BD 0000  A         JSR    SUB3
               *
045E 7D 0000  A         TST    F2
0461 27 06 0469         BEQ    A24
0463 7F 0000  A         CLR    FRSTQF
0466 7E 0000  A         JMP    RNG
               *
0469 7C 0000  A A24     INC    FRSTQF
046C CE 0000  A         LDX    #MES77   ZN UPPER LIMIT
046F BD 0000  A         JSR    PRINT
0472 86 01    A         LDAA   #1
0474 BD 0000  A         JSR    KBIN
0477 B6 0000  A         LDAA   BINHI
047A 84 7F    A         ANDA   #$7F
047C F6 0000  A         LDAB   BINLO
047F B7 0000  A         STAA   ZN
0482 F7 0001  A         STAB   ZN+1
0485 BD 0000  A         JSR    SUB3
               *
0488 7F 0000  A         CLR    FRSTOF
048B 7E 0000  A         JMP    RNG
               *
               *
048E CE 0000  A RENTR   LDX    #MES78
```

```
210   INIT   ***  INITIALIZATION SUBROUTINE  ***

 0491 BD 0000  A          JSR    PRINT
 0494 86 03    A          LDAA   #3
 0496 BD 0000  A          JSR    KBIN     FILLING TIME
 0499 B6 0000  A          LDAA   BINLO    (0-255 SEC MAX)
 049C 48                  ASLA            *2
 049D B7 0002  D          STAA   RSVFIL
 04A0 BD 0000  A          JSR    SUB3
                 *
 04A3 7D 0000  A          TST    F2       CHK IF CALLED FROM
 04A6 27 03 04AB          BEQ    A27      OTHER ROUTINE.
 04A8 7E 0000  A          JMP    RTNPT
                 *
 04AB CE 0000  A A27      LDX    #MES80
 04AE BD 0000  A          JSR    PRINT
 04B1 86 01    A          LDAA   #1       WANT MED EXCHG?
 04B3 BD 0000  A          JSR    KBIN
 04B6 B6 0000  A          LDAA   BINLO
 04B9 B7 0000  D          STAA   MEDEX
 04BC BD 0000  A          JSR    SUB3
                 *
 04BF 7D 0000  A          TST    F2
 04C2 27 03 04C7          BEQ    S2
 04C4 7E 0000  A          JMP    RTNPT
                 *
 04C7 7D 0000  D S2       TST    MEDEX    MED EXCHG?
 04CA 27 1D 04E9          BEQ    N1       NO: SKIP
                 *
 04CC CE 0000  A A28      LDX    #MES81   MED EXCHG FLUSHING
 04CF BD 0000  A          JSR    PRINT    TIME (0-120 SEC)
 04D2 86 02    A          LDAA   #2
 04D4 BD 0000  A          JSR    KBIN
 04D7 B6 0000  A          LDAA   BINLO
 04DA 48                  ASLA            (*2)
 04DB B7 0001  D          STAA   MEXCHG
 04DE BD 0000  A          JSR    SUB3
                 *
 04E1 7D 0000  A          TST    F2
 04E4 27 03 04E9          BEQ    N1
 04E6 7E 0000  A          JMP    RTNPT
                 *
 04E9 CE 0000  A N1       LDX    #MES12
 04EC C6 02    A          LDAB   #2       PRINT GO MESG
 04EE F7 0000  A          STAB   BLOCK
 04F1 BD 0000  A          JSR    SUB3
 04F4 BD 0000  A          JSR    SUB3
 04F7 BD 0000  A          JSR    PRINT1
 04FA BD 0000  A          JSR    SUB3
 04FD BD 0000  A          JSR    SUB3
                 *
                 *
 0500 7E 0000  A          JMP    RETSET   RETURN TO RESET
                 *
                          XREF   MES7,MES9,MES80,MES81
                          XREF   MES11,MES12,MES14,MES16,MES17
                          XREF   MES18,MES19,MES20,MES21,MES22,MES23
                          XREF   MES27,ACDFLG,MES40,MES10,MES24
                          XREF   MES25,MES26,MES78,SUB3,KBIN,FRSTQF
                          XREF   FRSTQ,BLOCK,PRINT1,PRINT
```

```
Ø11   INIT   ***  INITIALIZATION SUBROUTINE  ***
                              XREF    BCDHI,BCDLO,BINHI,BINLO,P4AP,P2BP
                              XREF    P2BC,RETSET,RCLFLG,MES65,CU,PB,CD
                              XREF    ZN,MES7Ø,MES71,MES72,MES73,MES74
                              XREF    MES75,MES76,MES77,RNG,F2,RTNPT
                         *
                         *
                              XDEF    SCANTH,SCANTL,STCN1H,STCN2H,RSVFIL
                              XDEF    STCN3H,STCN4H,VPHGH,VPHGL,MEDEX
                              XDEF    VPSAMH,VPSAML,VPVS1,VPVS2,MEXCHG
                              XDEF    VPFLSL,VPSTD,VPACID,VPPRGH,VPPRGL
                              XDEF    ZINK,VPRECY,INIT,RETINT,RENTR
                              XDEF    INTFLG,JØ,AØ,A1,A2,A3,A4,A5,A6,A27
                              XDEF    A7,A8,A9,A1Ø,A11,A12,A13,A14,A15,A2
                              XDEF    A16,A17,A18,A19,A2Ø,A21,A22,A23,A24
                         *
                         *
 ØØØØ                         DSCT
                         *
 ØØØØ      ØØØ1     A    MEDEX    RMB     1
 ØØØ1      ØØØ1     A    MEXCHG   RMB     1
 ØØØ2      ØØØ1     A    RSVFIL   RMB     1
 ØØØ3      ØØØ1     A    INTFLG   RMB     1
 ØØØ4      ØØØ1     A    SCANTH   RMB     1
 ØØØ5      ØØØ1     A    SCANTL   RMB     1
 ØØØ6      ØØØ1     A    VPHGH    RMB     1
 ØØØ7      ØØØ1     A    VPHGL    RMB     1
 ØØØ8      ØØØ1     A    VPSAMH   RMB     1
 ØØØ9      ØØØ1     A    VPSAML   RMB     1
 ØØØA      ØØØ1     A    VPVS1    RMB     1
 ØØØB      ØØØ1     A    VPVS2    RMB     1
 ØØØC      ØØØ1     A    VPFLSL   RMB     1
 ØØØD      ØØØ1     A    VPSTD    RMB     1
 ØØØE      ØØØ1     A    VPACID   RMB     1
 ØØØF      ØØØ1     A    VPPRGH   RMB     1
 ØØ1Ø      ØØØ1     A    VPPRGL   RMB     1
 ØØ11      ØØØ1     A    ZINK     RMB     1
 ØØ12      ØØØ1     A    VPRECY   RMB     1
                         *
 ØØØ8                         BSCT
                         *
 ØØØ8      ØØ3A     A    JØ       RMB     58
                         *
                              END
 ERRORS ØØØØØ
```

```
                        NAM    PROG     VER. 51  3-7-80   CLAVELL
                 *
                 *  FILE NAMES:   &PROG (S) / PROG (R) /
                 *
                        OPT    REL
                 *
                        TTL    ***  SYSTEM CONTROL PROGRAM  ***
                 *
                 ***********************************************
                 *
                 * THIS IS THE MAIN CONTROL PROGRAM FOR THE ASV
                 * INSTRUMENT SYSTEM FOLLOWING SYSTEM INITIALIZATI
                 *
                 ***********************************************
                 *
0000                    PSCT
                 *
0000    C0     A TTBL1  FCB    $C0,$79,$A4,$30,$99,$92,$03,$F8,$00
0001    79     A
0002    A4     A
0003    30     A
0004    99     A
0005    92     A
0006    03     A
0007    F8     A
0008    00     A
0009    18     A
                 *
                 *
0000                    DSCT
                 *
0000    07D0   A DATA   RMB    2000     STARTING ADDR. OF
                 *                 DATA BUFFER
                 *
000A                    PSCT
                 *
                 *    :::  MACRO DEFINITION  :::
                 *
                 ETM    MACR
                 *
                 \0 LDAB #$0F      LOAD COMAND
                  STAB P6BP
                  JSR CT
                  CLRA    DUMMY DIGIT
                  STAA P6BP
                  JSR CT
                  LDAA \1      LOAD BCD DIGITS
                  ANDA #$0F    MASK UPPER 4 BITS
                  STAA P5BP     STROBE IN LSD
                  JSR CT
                  LDAA \1      LOAD DIGITS AGAIN
                  LSRA       ROTATE RIGHT 4 TIMES
                  LSRA       TO GET 2 ND DIGIT
                  LSRA       INTO POSITION
                  LSRA
                  STAA P5BP  STROBE 2ND DIGIT
                  JSR CT
                  LDAA \2
                 ANDA #$ØF
                 STAA P6BP   3RD DIGIT
                 JSR CT
                 LDAA \2
                 LSRA
                 LSRA
                 LSRA
                 LSRA
                 STAA P5BP  STROBE 4TH DIGIT
                 JSR CT
                 LDAA #$ØØ    'PUT Ø'S IN DIGITS
                 STAA P5BP     5 & 6
                 JSR CT
                 JSR CT
                 ENDM
                *
                ETMS   MACR
                *
                 LDAA #$ØF        LOAD COMAND
                 STAA P6BP
                 JSR CT
                 LDAB #$ØØ         LOAD IN BCD DIGITS
                 STAB P6BP
                 JSR CT
                 JSR CT
                 LDAA \Ø
                 STAA P5BP
                 JSR CT
                 LDAA \1
                 STAA P5BP
                 JSR CT
                 STAB P6BP
                 JSR CT
                 JSR CT
                 JSR CT
                 ENDM
                *
                *
                *
ØØØA 7F ØØØØ  A PROGM  CLR    CNTR1    CLEAR FLAGS
ØØØD 7F ØØØØ  A        CLR    SCNFLG
ØØ1Ø 7F Ø7DE  D        CLR    SEC
ØØ13 7F Ø7DF  D        CLR    MIN
ØØ16 7F Ø7EØ  D        CLR    HR
ØØ19 7F ØØØØ  A        CLR    CSUMF
ØØ1C 7F ØØØØ  A        CLR    HALTF
ØØ1F 7F Ø7DC  D        CLR    MINØ
ØØ22 7F Ø7DD  D        CLR    HRØ
ØØ25 7F Ø7D1  D        CLR    PURGFG
ØØ28 7F Ø7DØ  D        CLR    DAYFG    DAY INC FLAG
                *
ØØ2B B6 ØØØØ  A        LDAA   P6AP     DUMMY READ
                *
ØØ2E CE ØØØØ  A        LDX    #TIMBUF  INITIALIZE POINTER
ØØ31 FF ØØØØ  A        STX    TMBUF    USED IN COMPT
                *
ØØ34 CE ØØØØ  D        LDX    #DATA
ØØ37 FF Ø7E1  D        STX    EPDATA   CALCULATE NEXT TO
```

```
                *                       LAST ADDR. IN DATA
003A F6 07E2  D          LDAB   EPDATA+1 BUFFER.  USED FOR
003D B6 07E1  D          LDAA   EPDATA   BUFFER OVER FLOW CHK.
                *
0040 CB C8    A          ADDB   #$C8     ADD 1992 TO DATA
0042 89 07    A          ADCA   #$07
                *
0044 B7 07E1  D          STAA   EPDATA
0047 F7 07E2  D          STAB   EPDATA+1
                *
004A 7D 0000  A          TST    RCLFLG
004D 27 2C 007B          BEQ    LEDCT
                *
004F BD 0000  A          JSR    CLOCK
                *
0052 B6 0000  A          LDAA   TBCDS    GET TIME OF FIRST
0055 B7 07DB  D          STAA   SEC0     SAMPLE
0058 B6 0000  A          LDAA   TBCDM
005B B7 07DC  D          STAA   MIN0
005E B6 0000  A          LDAA   TBCDH
0061 B7 07DD  D          STAA   HR0
                *
0064 7F 0000  A          CLR    P4AP
0067 7F 0000  A          CLR    P4BP
006A 86 10    A          LDAA   #$10     ACT V3A
006C B7 0000  A          STAA   P4AP
                *
006F 86 04    A          LDAA   #4
0071 BD 0000  A          JSR    GENTIM
                *
0074 86 60    A          LDAA   #$60     FILLING & SAMPLE
0076 B7 0000  A          STAA   P3AP     LED'S ON
0079 20 05 0080          BRA    FILL
                *
007B 86 A0    A LEDCT    LDAA   #$A0     FILLING & CONT.
007D B7 0000  A          STAA   P3AP     LED'S ON
                *
0080 86 01    A FILL     LDAA   #$01
0082 B7 0000  A          STAA   P4BP     FILL PUMP ON
                *
0085 B6 0000  A          LDAA   RSVFIL   FILLING TIME
0088 BD 0000  A          JSR    GENTIM
                *
008B 7F 0000  A          CLR    P4BP     FILL PUMP OFF
008E B6 0000  A          LDAA   P3AP
0091 84 C0    A          ANDA   #$C0     FILLING LED OFF
0093 B7 0000  A          STAA   P3AP
                *
                *
0096 B6 0000  A          LDAA   ACDFLG   ADD ACID?
0099 27 26 00C1          BEQ    SKIP3    NO: SKIP
                *
009B B6 0000  A          LDAA   P3AP     ACID PUMP ON
009E 8B 10    A          ADDA   #$10
00A0 B7 0000  A          STAA   P3AP
                *
00A3 B6 0000  A          LDAA   P3BP     LED ON
00A6 8B 40    A          ADDA   #$40
```

```
00A8 B7 0000   A          STAA   P3BP
                *
00AB B6 0000   A          LDAA   VPACID   GET ADDITION TIME
00AE BD 0000   A          JSR    GENTIM   TIME IT
                *
00B1 B6 0000   A          LDAA   P3BP
00B4 84 BF     A          ANDA   #$BF     LED OFF
00B6 B7 0000   A          STAA   P3BP
                *
00B9 B6 0000   A          LDAA   P3AP
00BC 84 EF     A          ANDA   #$EF     ACID PUMP OFF
00BE B7 0000   A          STAA   P3AP
                *
00C1 C6 02     A SKIP3    LDAB   #2       INITIALIZE DATA
00C3 F7 0000   A          STAB   CNTR2    VALID FLAG
                *
00C6 7D 07D1   D RTN      TST    PURGFG   PURGE DONE?
00C9 26 FB 00C6           BNE    RTN      NO : LOOP
                *
00CB 7D 0000   A          TST    HALTF    HALT FLAG SET?
00CE 27 03 00D3           BEQ    RTN1     NO: CONT
00D0 BD 0000   A          JSR    HALT     YES: JUMP TO HALT ROUT
                *
00D3 7F 07D7   D RTN1     CLR    VBFLG    CLR STD & ACID FLAGS
00D6 7F 07D6   D          CLR    VAFLG
00D9 7F 0000   A          CLR    P4AP
00DC B6 0000   A          LDAA   P4BP     CLR VALVES
00DF 84 03     A          ANDA   #$03
00E1 B7 0000   A          STAA   P4BP
                *
00E4 CE 07D0   A          LDX    #2000
00E7 FF 07E3   D          STX    CNTHLD   CLEAR DATA BUFFER
00EA CE 0000   D          LDX    #DATA
00ED FF 07D2   D          STX    TDATA
                *
00F0 4F                   CLRA            FILL WITH 0'S
00F1 FE 07D2   D DCLR     LDX    TDATA
00F4 A7 00     A          STAA   0,X
00F6 08                   INX
00F7 FF 07D2   D          STX    TDATA
00FA FE 07E3   D          LDX    CNTHLD
00FD 09                   DEX
00FE FF 07E3   D          STX    CNTHLD
0101 26 EE 00F1           BNE    DCLR     DONE ?
                *
0103 7F 07D9   D          CLR    TADU
0106 F6 0000   A          LDAB   CNTR1
0109 F7 07DA   D          STAB   TADL     GET CYCLE NO.
                *
010C FE 07D9   D          LDX    TADU     CONVERT TO 7 SEG.
010F A6 00     A          LDAA   TBL1,X   DISPLY CODE
                *
0111 B7 0000   A          STAA   P6AP     SET CYCLE LED
                *
0114 7D 0000   A          TST    CNTR1
0117 27 1E 0137           BEQ    SKIPX
                *
0119 B6 07DB   D          LDAA   SEC0
```

```
011C B7 07DE  D          STAA   SEC
011F B6 07DC  D          LDAA   MIN0     GET CURRENT SAMPLE
0122 B7 07DF  D          STAA   MIN      TIME FOR USE
0125 B6 07DD  D          LDAA   HR0      BY COMP
0128 B7 07E0  D          STAA   HR
                *
012B B6 0000  A          LDAA   DAYL0
012E B7 0000  A          STAA   DAYL
0131 B6 0000  A          LDAA   DAYH0
0134 B7 0000  A          STAA   DAYH
                *
0137 86 64    A SKIPX    LDAA   #$64     START MAIN PUMP
                *                         KEEP SCAN OFF
0139 B7 0000  A          STAA   P2BP     MET. PUMP OFF
013C 86 14    A          LDAA   #20      10 SEC DELAY LOOP
013E BD 0000  A          JSR    GENTIM
                *
0141 CE 0000  D          LDX    #DATA    GET STARTING ADR.
0144 FF 07D2  D          STX    TDATA    DATA BUFF & STOR
                *
0147 F6 0000  A          LDAB   IP10H    PUT IP UPPER INTO B
014A C4 03    A          ANDB   #$03     MASK UPPER 6 BITS
014C 86 E4    A          LDAA   #%11100100 LOAD A
014E 1B                  ABA             ADD A&B INTO A
014F F6 0000  A          LDAB   IP10L    LOWER 8 BITS INTO B
0152 B7 0000  A          STAA   P2BP     SET D/A OUTPUT
0155 F7 0000  A          STAB   P2AP     ELECTRODES ON
                *
0158 B6 0000  A          LDAA   P6AP     DUMMY READ/ZERO DETECT
                *
015B                     ETM    SKIP4,VPHGL,VPHGH SET E.T.-HG PLT.
                *
01A5 86 0A    A          LDAA   #$0A     SET HG LED
01A7 B7 0000  A          STAA   P3BP     ON=1
                *
01AA B6 0000  A SKIP5    LDAA   P6AC     TIME UP?
01AD 2A FB 01AA          BPL    SKIP6    NO-LOOP
                *
01AF B6 0000  A          LDAA   P6AP     DUMMY READ
                *
01B2 B6 0000  A          LDAA   OVR      CK OVR
01B5 84 01    A          ANDA   #$01
01B7 26 05 01BE          BNE    SKIP8
                *
01B9 86 02    A          LDAA   #$02     ACT V1B
01BB B7 0000  A          STAA   P4AP
                *
01BE B6 0000  A SKIP8    LDAA   VPVS1    VP DELAY
01C1 48                  ASLA            *2
01C2 BD 0000  A          JSR    GENTIM
                *
01C5 B6 0000  A          LDAA   OVR      CK OVR
01C8 85 02    A          BITA   #$02
01CA 26 08 01D4          BNE    SKIP9
                *                        OTHERWISE
01CC 7F 0000  A          CLR    P4AP     RESET
01CF 86 08    A          LDAA   #$08     & ACT V2B
01D1 B7 0000  A          STAA   P4AP
```

```
                    *
01D4 86 20      A SKIP9  LDAA   #$20      SAMPLE LED ON
01D6 B7 0000    A        STAA   P3BP      H3 & FULL LED'S OFF
                    *
                    *
01D9                     ETM    SKIP10,VPSAML,VPSAMH SAMP. TIME ON
                    *
                    *
0223 7D 0000    A        TST    RCLFLG    RECIRC MODE?
0226 27 0D 0235          BEQ    BRA6      NO:SKIP
                    *
0228 B6 0000    A        LDAA   P6AP      YES: DUMMY READ
022B B6 0000    A TIMUP  LDAA   P6AC      TIME UP ON E.T.?
022E 2A FB 022B          BPL    TIMUP     NO: LOOP
0230 B6 0000    A        LDAA   P6AP      DUMMY READ
                    *
0233 20 08 023D          BRA    SK12A
                    *
0235 B6 0000    A BRA6   LDAA   P1BP      DUMMY READ
                    *                          CLEAR RESV EMPTY FLG
0238 B6 0000    A SKIP12 LDAA   P1BC      CK FOR RESV EMPTY
023B 2A FB 0238          BPL    SKIP12    NO-LOOP
                    *
023D B6 0000    A SK12A  LDAA   P2BP
0240 84 FB      A        ANDA   #$FB
0242 B7 0000    A        STAA   P2BP      /PUMP OFF
                    *
0245 7F 0000    A        CLR    P3BP      SAMPLE PLATE LED OFF
                    *
0248 7D 0000    A        TST    MEDEX     USING MEDIUM EXCHG?
024B 27 5F 02AC          BEQ    SKIP7     NO: SKIP IT
                    *
                    ********************************************
                    *     MEDIUM EXCHANGE
                    ********************************************
                    *
024D B6 0000    A        LDAA   OVR
0250 85 08      A        BITA   #$08
0252 26 58 02AC          BNE    SKIP7     CK OVR
                    *
0254 7F 0000    A        CLR    P4AP      RESET TIMER
0257 7F 0000    A        CLR    P4BP
025A 86 04      A        LDAA   #$04      ACT V6A
025C B7 0000    A        STAA   P4BP
                    *
025F 86 02      A        LDAA   #2
0261 BD 0000    A        JSR    GENTIM
                    *
0264 7F 0000    A        CLR    P4BP
0267 86 01      A        LDAA   #$01      ACT V1A
0269 B7 0000    A        STAA   P4AP
                    *
026C 86 02      A        LDAA   #2
026E BD 0000    A        JSR    GENTIM
                    *
0271 7D 0000    A        TST    RCLFLG    V3 IN "A" POSIT?
0274 27 08 027E          BEQ    BRA3      NO:SKIP
                    *
```

```
0276 7F 0030  A        CLR    P4AP
0279 86 20    A        LDAA   #$20      ACT V3B
027B B7 0000  A        STAA   P4AP
                *
027E B6 0030  A BRA3   LDAA   P2BP
0281 8B 04    A        ADDA   #$04      PUMP ON
0283 B7 0030  A        STAA   P2BP
                *
0286 B6 0030  A        LDAA   MEXCHG    GET FLUSHING TIME
0289 BD 0000  A        JSR    GENTIM
                *
028C B6 0030  A        LDAA   P2BP
028F 84 FB    A        ANDA   #$FB      PUMP OFF
0291 B7 0030  A        STAA   P2BP
                *
0294 7F 0000  A        CLR    P4AP
0297 86 02    A        LDAA   #$02      ACT V1B
0299 B7 0000  A        STAA   P4AP
                *
029C 86 02    A        LDAA   #2
029E BD 0000  A        JSR    GENTIM
                *
02A1 7F 0000  A        CLR    P4BP
02A4 7F 0000  A        CLR    P4AP
02A7 86 08    A        LDAA   #$08      ACT V6B
02A9 B7 0000  A        STAA   P4BP
                *
02AC 86 1E    A SKIP7  LDAA   #30
02AE BD 0000  A        JSR    GENTIM    15 SEC EQUIL. TIME
                *
                *
                *
                **************************************************
                *            START SCAN
                **************************************************
                *
02B1 B6 0000  A SKIP13 LDAA   P2BP      PRESERVE 2 MSB
02B4 84 03    A        ANDA   #$03      OF D/A  /START CHART
02B6 C6 90    A        LDAB   #%10010000 PUT PEN DOWN
02B8 1B                ABA              START SCAN & ELECTRODES ON
02B9 B7 0030  A        STAA   P2BP
02BC B6 0030  A        LDAA   P6AP      DUMMY READ
                *
02BF                   ETM    BLANK,SCANTL,SCANTH SET ET TO
                *                                  SCAN TIME
0309 B6 0000  A        LDAA   CNTR2
030C 81 01    A        CMPA   #1        CNTR2 > 1
030E 2E 11 0321        BGT    SK16B     YES: SKIP
                *                        ELSE
0310 7F 0330  A        CLR    SCNFLG    CLR END OF SCAN FLG
0313 7C 07D8  D        INC    POLFLG    SET ENABLE SCAN FLAG
                *
                *
                ****     START TAKING DATA  ****
                *
0316 B6 0000  A        LDAA   P2AP      DUMMY READ
0319 86 07    A        LDAA   #07       CLR MASK ON CA1
031B B7 0000  A        STAA   P2AC      ENABLE INTERRUPTS FROM
```

```
031E 0E                       CLI               .2 SEC CLOCK INPUT
031F 20 05 0326               BRA    SKIP15
                      *
0321 86 07      A SK16B       LDAA   #$07       ENABLE ZERO INTERRUPT
0323 B7 0000    A             STAA   P6AC       ON E.T.
                      *
0326 7D 0000    A SKIP15      TST    RCLFLG
0329 27 0B 0336               BEQ    SK12
                      *
032B B6 0000    A             LDAA   CNTR1      LAST CYCLE?
032E B1 0000    A             CMPA   VPRECY
0331 27 03 0336               BEQ    SK12       YES: EMPTY RESV
                      *
0333 7E 03D6    P             JMP    SK1        NO: DON'T EMPTY
                      *
                      *
                      *
                      **********     FLUSH OUT RESV.    *************
                      *
0336 86 02      A SK12        LDAA   #$02
0338 B7 0000    A             STAA   P4BP       ACT V5B/START DRAINING
                      *
033B 86 10      A             LDAA   #$10       FLUSHING LED ON
033D B7 0000    A             STAA   P3BP
                      *
0340 7D 0000    A             TST    RCLFLG
0343 27 08 034D               BEQ    SK14A
                      *
0345 B6 0000    A             LDAA   P1BP       DUMMY READ
0348 B6 0000    A SK14C       LDAA   P1BC       RESV. EMPTY YET?
034B 2A FB 0348               BPL    SK14C      NO: LOOP
                      *                              YES: DO NORMAL FLUSH
                      *
034D 86 1E      A SK14A       LDAA   #30        15 SEC DELAY TO EMPTY
034F BD 0000    A             JSR    GENTIM     RESV. COMPLETELY
                      *
0352 C6 01      A             LDAB   #$01       STOP DRAINING
0354 F7 0000    A             STAB   P4BP       ACT V5A/FILL
                      *
0357 B6 0000    A             LDAA   OVR        CHK V5 OVR
035A 85 10      A             BITA   #$10       SKIP ALL FLUSHING
                      *                         TIMING IF V5
035C 27 03 0361               BEQ    SKIP17     IS IN OVERRIDE
                      *
035E 7E 03D6    P             JMP    SK1        NO FLUSH
                      *
                      *
0361 B6 0000    A SKIP17      LDAA   VPFLSL     FLUSH TIMING
0364 BD 0000    A             JSR    GENTIM     DO TIMING
                      *
                      *
0367 B6 0000    A SKIP20      LDAA   P1BP       DUMMY READ/CLR
                      *                         RESV. EMPTY FLAG
036A 86 02      A             LDAA   #$02       STOP FILLING
036C B7 0000    A             STAA   P4BP       ACT V5B/DRAIN
                      *
036F B6 0000    A SKIP22      LDAA   P1BC       RESV. EMPTY?
0372 2A FB 036F               BPL    SKIP22     YES-START FILLING
```

```
                 *                          OTHERWISE LOOP
                 *
0374 86 14      A SKIP23 LDAA    #20        10 SEC
0376 BD 0000    A        JSR     GENTIM     DELAY TO DRAIN COMPL
                 *
                 *
                 *
                 ***************************************************
                 *****    START RESERVOIR FILLING      ***********
                 ***************************************************
                 *
0379 86 01      A        LDAA    #$01       STOP DRAINING
037B B7 0000    A        STAA    P4BP       ACT V5A/FILL
                 *
037E 7F 0000    A        CLR     P3BP       FLUSHING LED OFF
                 *
0381 B6 0000    A        LDAA    P3AP       FILLING LED ON
0384 8B 20      A        ADDA    #$20
0386 B7 0000    A        STAA    P3AP
                 *
0389 BD 0000    A        JSR     CLOCK
                 *
038C B6 0000    A        LDAA    TBCDS
038F B7 07DB    D        STAA    SEC0
0392 B6 0000    A        LDAA    TBCDM      GET TIME OF DAY
0395 B7 07DC    D        STAA    MIN0       SAMPLE COLLECTED
0398 B6 0000    A        LDAA    TBCDH      FOR USE BY COMPT ROUTINE
039B B7 07DD    D        STAA    HR0
                 *
039E 84 F0      A        ANDA    #$F0       MASK LOWER BYTE
03A0 81 02      A        CMPA    #2         UPPER BYTE = 2?
03A2 26 09 03AD          BNE     SKIP25     NO: HR0<23
03A4 B6 07DD    D        LDAA    HR0        RESET A
03A7 84 0F      A        ANDA    #$0F       MASK UPPER BYTE
03A9 80 03      A        SUBA    #3         A - 3
03AB 2C 15 03C2          BGE     SKIP26     IF>= 0 THEN HR0 = 23,24
                 *                      NO: HR0 < 23
03AD 7D 07D0    D SKIP25 TST     DAYFG      DAY FLAG SET?
03B0 27 13 03C5          BEQ     SKIP28     NO: CONT.
                 *
03B2 7F 07D0    D        CLR     DAYFG      OTHER WISE CLEAR FLAG
03B5 7C 0000    A        INC     DAYL0      INC DAYL0
03B8 7D 0000    A        TST     DAYL0      DAYL0 = 0?
03BB 26 08 03C5          BNE     SKIP28     NO: CONT.
                 *
03BD 7C 0000    A        INC     DAYH0      OTHER WISE INC DAYH0
03C0 20 03 03C5          BRA     SKIP28     CONT.
                 *
03C2 7C 07D0    D SKIP26 INC     DAYFG
                 *
03C5 B6 0000    A SKIP28 LDAA    RSVFIL     GET FILL TIME
03C8 BD 0000    A        JSR     GENTIM     RESV. FILLING TIME
                 *
03CB B6 0000    A        LDAA    P3AP
03CE 84 C0      A        ANDA    #$C0       FILLING LED OFF
03D0 B7 0000    A        STAA    P3AP
                 *
03D3 7F 0000    A        CLR     P4BP       FILL PUMP OFF
```

```
                    *
                    ***********************************
                    *  SCANNING WAIT LOOP
                    ************************************
                    *
 Ø3D6 7D ØØØØ  A SK1    TST    SCNFLG    SCAN DONE?
 Ø3D9 27 FB Ø3D6        BEQ    SK1       NO: LOOP
                    *
                    *
 Ø3DB B6 ØØØØ  A SKIP36 LDAA   P2BP      TURN MAIN PUMP
 Ø3DE 8B Ø4    A        ADDA   #%ØØØØØ1ØØ & PUMP LED ON
 Ø3EØ B7 ØØØØ  A        STAA   P2BP
                    *
 Ø3E3 B6 ØØØØ  A        LDAA   OVR
 Ø3E6 85 Ø1    A        BITA   #$Ø1
 Ø3E8 26 21 Ø4ØB        BNE    SKIP37
                    *
 Ø3EA 7F ØØØØ  A        CLR    P4AP
 Ø3ED B6 ØØØØ  A        LDAA   P4BP
 Ø3FØ 84 Ø3    A        ANDA   #$Ø3
 Ø3F2 B7 ØØØØ  A        STAA   P4BP
 Ø3F5 86 Ø1    A        LDAA   #$Ø1      ACT V1A/
 Ø3F7 B7 ØØØØ  A        STAA   P4AP
                    *
 Ø3FA 7D ØØØØ  A        TST    MEDEX     ADD 1 SEC EXTRA
 Ø3FD 27 Ø5 Ø4Ø4        BEQ    VP1       DELAY IF DOING MED EX
                    *
 Ø3FF 86 Ø2    A        LDAA   #2
 Ø4Ø1 BD ØØØØ  A        JSR    GENTIM
                    *
 Ø4Ø4 B6 ØØØØ  A VP1    LDAA   VPVS2     GET VP DELAY 2
 Ø4Ø7 48                ASLA             *2
 Ø4Ø8 BD ØØØØ  A        JSR    GENTIM
                    *
 Ø4ØB B6 ØØØØ  A SKIP37 LDAA   OVR
 Ø4ØE 85 Ø2    A        BITA   #$Ø2
 Ø41Ø 26 33 Ø445        BNE    SKIP38
                    *
 Ø412 B6 ØØØØ  A        LDAA   P4BP
 Ø415 84 Ø3    A        ANDA   #$Ø3
 Ø417 B7 ØØØØ  A        STAA   P4BP
 Ø41A 7F ØØØØ  A        CLR    P4AP
 Ø41D 86 Ø4    A        LDAA   #$Ø4
 Ø41F B7 ØØØØ  A        STAA   P4AP      ACT V2A/HG LOOP
                    *
 Ø422 86 Ø2    A        LDAA   #2
 Ø424 BD ØØØØ  A        JSR    GENTIM
                    *
 Ø427 7D ØØØØ  A        TST    MEDEX
 Ø42A 27 19 Ø445        BEQ    SKIP38
                    *
 Ø42C 7D ØØØØ  A        TST    RCLFLG    RECIRC MODE?
 Ø42F 27 14 Ø445        BEQ    SKIP38    NO: SKIP
                    *
 Ø431 B6 ØØØØ  A        LDAA   OVR       YES: RESET V3
 Ø434 85 Ø4    A        BITA   #$Ø4
 Ø436 26 ØD Ø445        BNE    SKIP38
                    *
```

```
0438 7F 0000  A        CLR    P4AP
043B 86 10    A        LDAA   #$10      ACT V3A
043D B7 0000  A        STAA   P4AP      RECIRC MODE
                *
0440 86 02    A        LDAA   #2
0442 BD 0000  A        JSR    GENTIM
                *
0445 B6 0000  A SKIP38 LDAA   CNTR2     CNTR2=0?
0448 27 03 044D        BEQ    SKIP39
044A 7A 0000  A        DEC    CNTR2     NO : DEC CNTR2
                *
                *
044D B6 0000  A SKIP39 LDAA   P6AP      DUMMY READ/CLEAR
                *                     E.T. ZERO
0450 7D 0000  A        TST    RCLFLG
0453 27 0B 0460        BEQ    PURG
                *
0455 B6 0000  A        LDAA   CNTR1
0458 B1 0000  A        CMPA   VPRECY
045B 27 03 0460        BEQ    PURG      NEW SAMPLE- PURGE
045D 7E 04BA  P        JMP    NOPURG    SAME SAMPLE - NO PURGE
                *
                *
0460                   ETM    PURG,VPPRGL,VPPRGH PURGE TM-E.T.
                *
04AA B6 0000  A        LDAA   P3BP      PURGE LED ON
04AD 8B 04    A        ADDA   #$04
04AF B7 0000  A        STAA   P3BP
                *
04B2 86 07    A        LDAA   #$07      ENABLE E.T. 0
04B4 B7 0000  A        STAA   P6AC      INTERRUPT
                *
04B7 7C 07D1  D        INC    PURGFG
                *                     DUMMY READ TO
04BA B6 0000  A NOPURG LDAA   P1BP      CLR RESV. EMP. FLAG
                *
04BD 7D 0000  A        TST    RCLFLG
04C0 27 0A 04CC        BEQ    SK25A
                *
04C2 B6 0000  A        LDAA   CNTR1
04C5 B1 0000  A        CMPA   VPRECY    LAST CYCLE ?
04C8 27 02 04CC        BEQ    SK25A     YES: SKIP
                *
04CA 20 2B 04F7        BRA    SKIP27    NO: NO ACID ADDITION
                *
04CC B6 0000  A SK25A  LDAA   ACDFLG    MAKE ACID ADD.?
04CF 27 26 04F7        BEQ    SKIP27    NO-SKIP
                *
04D1 B6 0000  A        LDAA   P3AP      ACID PUMP ON
04D4 8B 10    A        ADDA   #$10
04D6 B7 0000  A        STAA   P3AP
                *
04D9 B6 0000  A        LDAA   P3BP
04DC 8B 40    A        ADDA   #$40      ACID ADD LED ON
04DE B7 0000  A        STAA   P3BP
                *
04E1 B6 0000  A        LDAA   VPACID    ACID ADD TIME
04E4 BD 0000  A        JSR    GENTIM
```

```
                  *
04E7 B6 0000    A          LDAA   P3AP     ACID PUMP OFF
04EA 84 EF      A          ANDA   #$EF
04EC B7 0000    A          STAA   P3AP
                  *
04EF B6 0000    A          LDAA   P3BP
04F2 84 BF      A          ANDA   #$BF     ACID LED OFF
04F4 B7 0000    A          STAA   P3BP
                  *
                  *
04F7 B6 0000    A  SKIP27  LDAA   CNTR1    TIME TO ADD STD?
04FA 8B 01      A          ADDA   #1       ADD 1 FOR LOOK AHEAD
04FC B1 0000    A          CMPA   VPRECY   VALUE AND COMPARE
04FF 26 26 0527            BNE    SKIP32   NO STD ADD-SKIP
                  *                      OTHERWISE CONT.
0501 B6 0000    A          LDAA   P2BP
0504 8B 08      A          ADDA   #$08     STD PUMP ON
0506 B7 0000    A          STAA   P2BP
                  *
0509 B6 0000    A          LDAA   P3BP
050C 8B 80      A          ADDA   #$80
050E B7 0000    A          STAA   P3BP     STD ADD LED ON
                  *
                  *
0511 B6 0000    A          LDAA   VPSTD    STD ADD TIME
0514 BD 0000    A          JSR    GENTIM
                  *
0517 B6 0000    A          LDAA   P2BP
051A 84 F7      A          ANDA   #$F7     STD PUMP OFF
051C B7 0000    A          STAA   P2BP
                  *
051F B6 0000    A          LDAA   P3BP
0522 84 7F      A          ANDA   #$7F     LED OFF
0524 B7 0000    A          STAA   P3BP
                  *
0527 BD 0000    A  SKIP32  JSR    COMPT    COMPUTE ROUTINE
                  *
052A 7C 0000    A          INC    CNTR1
                  *
052D B6 0000    A          LDAA   VPRECY   GET STD ADD CYCLE #
0530 8B 01      A          ADDA   #1       INC BY 1
0532 B1 0000    A          CMPA   CNTR1    COMP TO CNTR1
0535 27 03 053A            BEQ    SKIP43   IF=,CLR CNTR1 & RTN
0537 7E 00C6    P          JMP    RTN
                  *
053A 7F 0000    A  SKIP43  CLR    CNTR1
053D 7C 0000    A          INC    CNTR1
0540 7E 00C6    P          JMP    RTN
                  *
                  *
                  *
                  *
                           XREF   GENTIM,TBCDS,TBCDM,TBCDH,CNTR1
                           XREF   SCNFL3,CNTR2,MEDEX,MEXCHG
                           XREF   IP10L,VPVS1,VPSAML,VPSAMH,COMPT
                           XREF   SCAN,CLOCK,VPSTD,RSVFIL
                           XREF   BCDBIN,ACDFLG,VPRECY,DAYL,DAYH
                           XREF   VPPRGH,VPPRGL,P1AP,DAYL0,DAYH0
```

```
Ø13   PROG   ***  SYSTEM CONTROL PROGRAM  ***

                                XREF   P1BP,P1AC,P1BC,P2AP,P2BP,P2AC,P2BC
                                XREF   P3AP,P3BP,P3AC,P3BC,P4AP,P4BP,P4AC
                                XREF   P4BC,P5AP,P6BP,P6AC,P6BC,HALT
                                XREF   VPHGH,VPHGL,IP1ØH,CSUMF,HALTF
                                XREF   RCLFLG,SCANTH,SCANTL,VPFLSL,CT,OVR
                                XREF   TIMBUF,TMBUF,VPACID,TBL1,VPVS2
                          *
                          *
                          *
                                XDEF   DATA,TDATA,PROGM,SEC,MIN,HR,CNTHLD
                                XDEF   VAFLG,VBFLG,POLFLG,EPDATA
                                XDEF   HRØ,MINØ,SECØ,TTBL1,PURGFG
                          *
                          *
 Ø7DØ                           DSCT
                          *
 Ø7DØ      ØØØ1    A  DAYFG    RMB      1
 Ø7D1      ØØØ1    A  PURGFG   RMB      1
 Ø7D2      ØØØ2    A  TDATA    RMB      2
 Ø7D4      ØØØ1    A  BCDMS    RMB      1
 Ø7D5      ØØØ1    A  BCDLS    RMB      1
 Ø7D6      ØØØ1    A  VAFLG    RMB      1
 Ø7D7      ØØØ1    A  VBFLG    RMB      1
 Ø7D8      ØØØ1    A  POLFLG   RMB      1
 Ø7D9      ØØØ1    A  TADU     RMB      1
 Ø7DA      ØØØ1    A  TADL     RMB      1
 Ø7DB      ØØØ1    A  SECØ     RMB      1
 Ø7DC      ØØØ1    A  MINØ     RMB      1
 Ø7DD      ØØØ1    A  HRØ      RMB      1
 Ø7DE      ØØØ1    A  SEC      RMB      1
 Ø7DF      ØØØ1    A  MIN      RMB      1
 Ø7EØ      ØØØ1    A  HR       RMB      1
 Ø7E1      ØØØ2    A  EPDATA   RMB      2
 Ø7E3      ØØØ2    A  CNTHLD   RMB      2
                          *
                          *
                                END
ERRORS ØØØØØ
```

```
 NAM MESG     VER. 17      9-19-79    CLAVELL
*
*   FILE NAMES:  &MESG (S) / MESG (R)
*
 OPT REL
*
 TTL    ***  MESSAGE LIST SUBROUTINE  ***
*
**************************************************************
*
*   THIS IS A LIST OF ALL THE MESSAGES CALLED BY OTHER
*   SYSTEM PROGRAMES.
*
****************************************************************
*
*
 PSCT

     FCC / YEAR:                                  /
     FCC / JULIAN DATE:                           /
     FCC /SET REAL-TIME CLOCK TO CURRENT TIME:    /
     FCC /    THEN PUSH "GO"                      /

     FCC /13 CD STANDARD CONC.-IN PPB:            /
     FCC /14 PB STANDARD CONC.-IN PPB:            /
      FCC /15 CU STANDARD CONC.-IN PPB:            /
      FCC /16 ZN STANDARD CONC.-IN PPB:            /

      FCC / IF THE POTENTIOSTATE PARAMETERS        /
      FCC /   ARE SET, PUSH  "GO"                  /

      FCC /  * * * * * * * * * * * * * * * * * *   /
      FCC /  *    SYSTEM INITIALIZATION        *   /
      FCC /  *   ENTER THE FOLLOWING DATA      *   /
      FCC /  * * * * * * * * * * * * * * * * * *   /

      FCC /Ø HG PLATTING TIME (MIN&SEC):           /
      FCC /1 SAMPLE PLATE TIME(MIN&SEC):           /
      FCC /2 VALVE DELAY1 (SEC):                   /
      FCC /3 VALVE DELAY2 (SEC):                   /
      FCC /4 SCAN TIME (MIN&SEC):                  /
      FCC /5 FLUSHING TIME (SEC):                  /
      FCC /6 STD. ADDITION TIME (SEC):             /
      FCC /8 ACID ADDITION TIME (SEC):             /
      FCC /9 PURGE DELAY TIME (MIN&SEC):           /
      FCC /1Ø ZINC ANALYSIS (YES=1,NO=Ø):          /
      FCC /11 RECIRC. MODE (YES=1,NO=Ø):           /
      FCC /12 CYCLE # FOR STD. ADD .:              /
      FCC /25 INITIAL POTENTIAL:                   /
      FCC /FINAL POTENTIAL:                        /
 FCC /INPUT ERROR: PLEASE REENTER NUMBER        /
FCC /***  SELECT PROGRAM MODE  ***              /
FCC /7 ACID ? (YES=1,NO=Ø):                     /
FCC /DATA OUTPUT  ** CONCENTRATIONS IN PPB **/
FCC /*** ERROR: LOW STD CONC-   ,  ,     ***    /
FCC /PARAMETER CHANGE (YES=1,NO=Ø):             /
FCC /PUSH "GO" TO RESTART PROGRAM               /
FCC /ERROR: NUMBER MUST BE > 1 & < 12Ø          /
FCC /17 CD LOWER LIMIT (IN VOLTS):              /
FCC /18 CD UPPER LIMIT (IN VOLTS):              /
FCC /19 PB LOWER LIMIT (IN VOLTS):              /
FCC /2Ø PB UPPER LIMIT (IN VOLTS):              /
FCC /21 CU LOWER LIMIT (IN VOLTS):              /
FCC /22 CU UPPER LIMIT (IN VOLTS):              /
FCC /23 ZN LOWER LIMIT (IN VOLTS):              /
FCC /24 ZN UPPER LIMIT (IN VOLTS):              /
FCC /26 RESV FILLING TIME (SEC):                /
FCC /QUESTION NUMBER:                           /
FCC /27 MEDIUM EXCHG (YES=1,NO=Ø):              /
FCC /28 MEDIUM FLUSH TIME (SEC):                /


MES1,MES2,MES3,MES7,MES9,MES1Ø,MES8Ø,MES81
MES11,MES12,MES14,MES15,MES17,MES18,MES19
MES2Ø,MES21,MES22,MES23,MES24,MES25,MES26,MES27
MES3Ø,MES31,ERRMES,MES32,MES4Ø,MES45,MESER
MES65,MES7Ø,MES71,MES72,MES73,MES78
MES74,MES75,MES76,MES77,MES5Ø,MES6Ø,MES79
```

```
                        NAM    POLL     VER. 16  8-2-79  CLAVELL
                        OPT    REL
                   *
                   *  FILE NAMES:   &POL (S) / POL (R)
                   *
                   ************************************************
                        TTL    *** INTERRUPT POLLING ROUTINE ***
                   ************************************************
                   *
                   *
                   *
                   *******    INTERRRUPT CLOCK INTERVAL IS .2 SEC
                   *
0000                    PSCT
                   *
0000 B6 0000  A POLL    LDAA   POLFLG    SCANNING FLAG SET?
0003 27 08 000D         BEQ    POL       NO: GO TO POL
                   *
0005 B6 0000  A         LDAA   P2AC      SCAN INTERRUPT?
0008 2A 03 000D         BPL    POL       NO:  JMP TO POL
000A 7E 010C  P         JMP    POL3      YES: JMP TO SCAN
                   *
000D B6 0000  A POL     LDAA   P6AC      E.T. ZERO?
0010 2A 06 0018         BPL    HLT       NO: GO TO HLT
0012 7E 0152  P         JMP    POL17     YES:GO TO POL17
                   *
0015 7E 0124  P POLB    JMP    POL2
                   *
0018 B6 0000  A HLT     LDAA   P5BC      HALT FLAG SET?
001B 48                 ASLA
001C 2A 03 0021         BPL    VAL       NO: CHK OVR'S
001E 7E 013E  P         JMP    POL16     YES: JMP TO HALT
                   *
0021 B6 0000  A VAL     LDAA   P2BC      OVR-RESET SET?
0024 2B EF 0015         BMI    POLB
                   *
0026 C6 28    A         LDAB   #40
0028 CE 0000  A         LDX    #PRBUF    MOVE MESG INTO
002B FF 0000  A         STX    TMES      RAM PRINT BUFFER
002E CE 01AB  P         LDX    #MESOVR
0031 FF 0000  A         STX    T1MES
                   *
0034 FE 0000  A         LDX    T1MES
0037 A6 00    A BOOT1   LDAA   0,X
0039 FE 0000  A         LDX    TMES
003C A7 00    A         STAA   0,X
003E 08                 INX
003F FF 0000  A         STX    TMES      UPDATE POINTER
0042 FE 0000  A         LDX    T1MES
0045 08                 INX
0046 FF 0000  A         STX    T1MES     UPDATE POINTER
0049 5A                 DECB             DEC COUNTER
004A 26 EB 0037         BNE    BOOT1
                   *
004C B6 0000  A         LDAA   P3AC
004F 2B 1B 006C         BMI    POL4      V1?
                   *
0051 48                 ASLA
```

```
0052 2B 28 007C             BMI    POL5      V2?
                   *
0054 B6 0000  A             LDAA   P3BC
0057 2B 33 008C             BMI    POL6      V3?
                   *
0059 48                     ASLA
005A 2B 40 009C             BMI    POL7      V4?
                   *
005C B6 0000  A             LDAA   P4AC
005F 2B 4B 00AC             BMI    POL8      V5?
                   *
0061 48                     ASLA
0062 2B 58 00BC             BMI    POL9      V6?
                   *
0064 B6 0000  A             LDAA   P4BC
0067 2B 63 00CC             BMI    POL10     V7?
                   *
0069 48                     ASLA
006A 2B 70 00DC             BMI    POL11     V8?
                   *
                   *
006C B6 0001  D    POL4     LDAA   OVR       TEST INTERRUPT REG.
006F 8B 01    A             ADDA   #$01
0071 B7 0001  D             STAA   OVR
0074 C6 31    A             LDAB   #$31
0076 F7 0006  A             STAB   PRBUF+6
0079 7E 00EC  P             JMP    BACK
                   *
007C B6 0001  D    POL5     LDAA   OVR
007F 8B 02    A             ADDA   #$02
0081 B7 0001  D             STAA   OVR
0084 C6 32    A             LDAB   #$32
0086 F7 0006  A             STAB   PRBUF+6
0089 7E 00EC  P             JMP    BACK
                   *
008C B6 0001  D    POL6     LDAA   OVR
008F 8B 04    A             ADDA   #$04
0091 B7 0001  D             STAA   OVR
0094 C6 33    A             LDAB   #$33
0096 F7 0006  A             STAB   PRBUF+6
0099 7E 00EC  P             JMP    BACK
                   *
009C B6 0001  D    POL7     LDAA   OVR
009F 8B 08    A             ADDA   #$08
00A1 B7 0001  D             STAA   OVR
00A4 C6 34    A             LDAB   #$34
00A6 F7 0006  A             STAB   PRBUF+6
00A9 7E 00EC  P             JMP    BACK
                   *
00AC B6 0001  D    POL8     LDAA   OVR
00AF 8B 10    A             ADDA   #$10
00B1 B7 0001  D             STAA   OVR
00B4 C6 35    A             LDAB   #$35
00B6 F7 0006  A             STAB   PRBUF+6
00B9 7E 00EC  P             JMP    BACK
                   *
00BC B6 0001  D    POL9     LDAA   OVR
00BF 8B 20    A             ADDA   #$20
```

```
003   POLL   *** INTERRUPT POLLING ROUTINE ***

 00C1 B7 0001  D         STAA   OVR
 00C4 C6 36    A         LDAB   #$36
 00C6 F7 0006  A         STAB   PRBUF+6
 00C9 7E 00EC  P         JMP    BACK
                 *
 00CC B6 0001  D POL10   LDAA   OVR
 00CF 8B 40    A         ADDA   #$40
 00D1 B7 0001  D         STAA   OVR
 00D4 C6 37    A         LDAB   #$37
 00D6 F7 0006  A         STAB   PRBUF+6
 00D9 7E 00EC  P         JMP    BACK
                 *
 00DC B6 0001  D POL11   LDAA   OVR
 00DF 8B 80    A         ADDA   #$80
 00E1 B7 0001  D         STAA   OVR
 00E4 C6 38    A         LDAB   #$38
 00E6 F7 0006  A         STAB   PRBUF+6
 00E9 7E 00EC  P         JMP    BACK
                 *
                 *
 00EC 86 01    A BACK    LDAA   #1          OVER RIDE MESS.
 00EE 97 07    A         STAA   $0007
                 *
 00F0 CE 0000  A         LDX    #PRBUF
 00F3 DF 05    A         STX    $0005
                 *
 00F5 BD B2C3  A         JSR    $B2C3
 00F8 BD B2A0  A         JSR    $B2A0
 00FB BD B38B  A         JSR    $B38B
                 *
 00FE B6 0000  A         LDAA   P3AP        DUMMY READS TO
 0101 B6 0000  A         LDAA   P3BP        CLEAR INTERRUPT
 0104 B6 0000  A         LDAA   P4AP        FLAGS
 0107 B6 0000  A         LDAA   P4BP
                 *
 010A 0E                 CLI
 010B 3B                 RTI
                 *
 010C 7C 0000  D POL3    INC    CSUMF       ACCUMULATE 3 INTERRUPTS
 010F B6 0000  D         LDAA   CSUMF       BEFORE COLLECTING
 0112 81 03    A         CMPA   #3          DATA POINTS
 0114 26 06 011C         BNE    RPO1        .6 SEC/PT
                 *
 0116 7E 0000  A         JMP    SCAN
 0119 7F 0000  D RPOL    CLR    CSUMF
                 *
 011C B6 0000  A RPO1    LDAA   P2BP        DUMMY READ
 011F B6 0000  A         LDAA   P2AP        DUMMY READ
                 *
 0122 0E                 CLI
 0123 3B                 RTI
                 *
 0124 7F 0001  D POL2    CLR    OVR         CLR INTR. REGISTER
                 *
 0127 86 01    A         LDAA   #1
 0129 97 07    A         STAA   $0007
                 *
 012B CE 0183  P         LDX    #MESP       POLLING MESG.
```

```
ØØ4   POLL   *** INTERRUPT POLLING ROUTINE ***

Ø12E DF Ø5       A         STX    $ØØØ5
                   *
Ø13Ø BD B2C3     A         JSR    $B2C3
Ø133 BD B2AØ     A         JSR    $B2AØ
Ø136 BD B38B     A         JSR    $B38B
                   *
Ø139 B6 ØØØØ     A         LDAA   P2BP       DUMMY READ
                   *
Ø13C ØE                    CLI
Ø13D 3B                    RTI
                   *
Ø13E 7D ØØØØ     A POL15   TST    INTFLG     CALLED FROM INIT?
Ø141 27 Ø5 Ø148            BEQ    HLT1       NO
Ø143 BD ØØØØ     A         JSR    CHG        YES: JUMP TO HALT
                   *
Ø146 ØE                    CLI
Ø147 3B                    RTI
                   *
Ø148 86 Ø1       A HLT1    LDAA   #1         SET THE HALT FLAG
Ø14A B7 ØØØ2     D         STAA   HALTF
Ø14D B6 ØØØØ     A         LDAA   P5BP       DUMMY READ
Ø15Ø ØE                    CLI
Ø151 3B                    RTI
                   *
Ø152 86 Ø6       A POL17   LDAA   #$Ø6       MASK INTERRUPT
Ø154 B7 ØØØØ     A         STAA   P6AC       ON E.T. ZERO
                   *
Ø157 7D ØØØØ     A         TST    PURGFG     FROM PURGE?
Ø15A 27 ØD Ø169            BEQ    SCN        NO: SCAN
                   *
Ø15C B6 ØØØØ     A         LDAA   P3BP
Ø15F 84 FB       A         ANDA   #$FB       PURGE LED OFF
Ø161 B7 ØØØØ     A         STAA   P3BP
Ø164 7F ØØØØ     A         CLR    PURGFG
Ø167 2Ø 15 Ø17E            BRA    SCN1
                   *
Ø169 7C ØØØØ     A SCN     INC    SCNFLG     SET SCAN END FLAG
Ø16C 7F ØØØØ     A         CLR    POLFLG
Ø16F 86 Ø4       A         LDAA   #$Ø4
Ø171 B7 ØØØØ     A         STAA   P2AC
                   *
Ø174 B6 ØØØØ     A         LDAA   P2BP
Ø177 84 Ø3       A         ANDA   #$Ø3       END SCAN
Ø179 8B 6Ø       A         ADDA   #$6Ø
Ø17B B7 ØØØØ     A         STAA   P2BP
                   *
Ø17E B6 ØØØØ     A SCN1    LDAA   P6AP       DUMMY READ
Ø181 ØE                    CLI
Ø182 3B                    RTI
                   *
                   *
                           XREF   SCAN,P2AC,P2BC,P3AP,P3BP
                           XREF   P2BP,POLFLG,P2AP,P3AC,P3BC
                           XREF   P4AP,P4BP,P4AC,P4BC,P5BC
                           XREF   PRBUF,TMES,T1MES~~,PRXFG,~~P5BP
                           XREF   SCNFLG,P6AC,P6AP,PURGFG,INTFLG,CHG
                   *
                           XDEF   POLL,RPOL,OVR,CSUMF,HALTF
```

```
                      *
 Ø183     4F      A MESP    FCC     /OVER-RIDE FLAGS RESET
 Ø184     56      A
 Ø185     45      A
 Ø186     52      A
 Ø187     2D      A
 Ø188     52      A
 Ø189     49      A
 Ø18A     44      A
 Ø18B     45      A
 Ø18C     2Ø      A
 Ø18D     46      A
 Ø18E     4C      A
 Ø18F     41      A
 Ø19Ø     47      A
 Ø191     53      A
 Ø192     2Ø      A
 Ø193     52      A
 Ø194     45      A
 Ø195     53      A
 Ø196     45      A
 Ø197     54      A
 Ø198     2Ø      A
 Ø199     2Ø      A
 Ø19A     2Ø      A
 Ø19B     2Ø      A
 Ø19C     2Ø      A
 Ø19D     2Ø      A
 Ø19E     2Ø      A
 Ø19F     2Ø      A
 Ø1AØ     2Ø      A
 Ø1A1     2Ø      A
 Ø1A2     2Ø      A
 Ø1A3     2Ø      A
 Ø1A4     2Ø      A
 Ø1A5     2Ø      A
 Ø1A6     2Ø      A
 Ø1A7     2Ø      A
 Ø1A8     2Ø      A
 Ø1A9     2Ø      A
 Ø1AA     2Ø      A
                      *
 Ø1AB     56      A MESOVR FCC      /VALVE    OVER-RIDE SET.
 Ø1AC     41      A
 Ø1AD     4C      A
 Ø1AE     56      A
 Ø1AF     45      A
 Ø1BØ     2Ø      A
 Ø1B1     2Ø      A
 Ø1B2     2Ø      A
 Ø1B3     4F      A
 Ø1B4     56      A
 Ø1B5     45      A
 Ø1B6     52      A
 Ø1B7     2D      A
 Ø1B8     52      A
 Ø1B9     49      A
 Ø1BA     44      A
```

```
ØØ6   POLL   *** INTERRUPT POLLING ROUTINE ***

 Ø1BB    45    A
 Ø1BC    2Ø    A
 Ø1BD    53    A
 Ø1BE    45    A
 Ø1BF    54    A
 Ø1CØ    2E    A
 Ø1C1    2Ø    A
 Ø1C2    2Ø    A
 Ø1C3    2Ø    A
 Ø1C4    2Ø    A
 Ø1C5    2Ø    A
 Ø1C6    2Ø    A
 Ø1C7    2Ø    A
 Ø1C8    2Ø    A
 Ø1C9    2Ø    A
 Ø1CA    2Ø    A
 Ø1CB    2Ø    A
 Ø1CC    2Ø    A
 Ø1CD    2Ø    A
 Ø1CE    2Ø    A
 Ø1CF    2Ø    A
 Ø1DØ    2Ø    A
 Ø1D1    2Ø    A
 Ø1D2    2Ø    A
                  *
 ØØØØ                     DSCT
                  *
 ØØØØ     ØØØ1   A CSUMF  RMB     1
 ØØØ1     ØØØ1   A OVR    RMB     1
 ØØØ2     ØØØ1   A HALTF  RMB     1
                  *
                          END
ERRORS ØØØØØ
```

```
                                  NAM     SCAN     VER. 13   4-15-79    CLAVE
                          *
                          *    FILE NAMES:    &SCAN / SCAN (R)
                          *
                                  OPT     REL
                          *
                                  TTL     *** DATA COLLECTING ROUTINE ***
                          *
                          ****************************************
                          *
                          * ROUTINE TO COLLECT AND STORE "Y"
                          * DATA VALUES FOR USE BY COMPUT
                          * SUBROUTINE.  (1 X-Y VALUE PAIR / .6 SEC)
                          *
                          ****************************************
                          *
                          *
0000                              PSCT
                          *
0000 FE 0000   A SCAN     LDX     EPDATA    CHK NEXT TO LAST DATA BUFF
0003 A6 00     A          LDAA    X         ADDRESS FOR 0'S,
0005 81 00     A          CMPA    #0        IF NOT 0 - BUFF OVER-
0007 27 03 000C           BEQ     OVRF      FLOW
0009 7E 008C   P          JMP     OVRFLO    GIVE OVER FLOW MESS
                          *
000C CE 000A   A OVRF     LDX     #10
000F FF 0000   D          STX     DTEN
                          *
0012 86 20     A          LDAA    #$20      Y AXIS/SET BIT 5
0014 B7 0000   A          STAA    P1BP
                          *
0017 86 3C     A          LDAA    #$3C      START A/D CONV.
0019 B7 0000   A          STAA    P1AC
001C 01                   NOP
001D 01                   NOP
001E 86 34     A          LDAA    #$34      RESET BIT
0020 B7 0000   A          STAA    P1AC
0023 B6 0000   A CONV     LDAA    P1AC      CHECK FOR CONV.
0026 2A FB 0023           BPL     CONV      COMPLETE-BIT 7 SET
                          *
0028 86 A7     A          LDAA    #167
002A 4A          D1       DECA              1 MS DELAY
002B 26 FD 002A           BNE     D1
                          *
002D B6 0000   A          LDAA    P1BP      GET UPPER BYTE OF Y
0030 84 0F     A          ANDA    #$0F      MASK UPPER 4 BITS
0032 F6 0000   A          LDAB    P1AP      GET LSB OF Y
                          *
                          ************************
                          *  DIVIDE Y VALUE BY 10
                          ************************
0035 CE 0000   D          LDX     #DTEN     GET ADDR OF HI
0038 BD 0000   A          JSR     DIV16     BYTE OF DIVISOR
                          *
003B FE 0000   A          LDX     TDATA     STORE 1ST Y VALUE
003E A7 00     A          STAA    X         MSB
0040 08                   INX
0041 E7 00     A          STAB    X         LSB
```

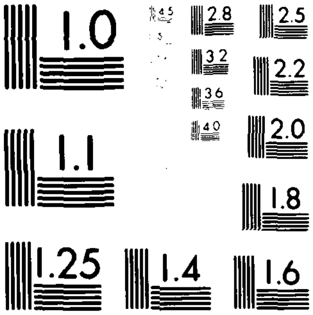
```
0043 08                 INX               INCREMENT POINTER
0044 FF 0000   A        STX     TDATA     SAVE NEW ADDRESS
                 *
0047 FE 0002   D        LDX     NUMBYH    INC Y COUNTER
004A 08                 INX
004B FF 0002   D        STX     NUMBYH
                 *
004E 4F                 CLRA              X AXIS
004F B7 0000   A        STAA    P1BP
                 *
0052 86 3C     A        LDAA    #$3C      START A/D CONV.
0054 B7 0000   A        STAA    P1AC
0057 01                 NOP
0058 01                 NOP
0059 86 34     A        LDAA    #$34      RESET BIT
005B B7 0000   A        STAA    P1AC
005E B6 0000   A CONV1  LDAA    P1AC      CHECK FOR CONV.
0061 2A FB 005E         BPL     CONV1     COMPLETE-BIT 7 SET
                 *
0063 86 A7     A        LDAA    #167
0065 4A          D2     DECA
0066 26 FD 0065         BNE     D2
                 *
0068 B6 0000   A        LDAA    P1BP      GET MSB OF X
006B 84 0F     A        ANDA    #$0F      MASK UPPER 4 BITS
006D F6 0000   A        LDAB    P1AP      GET LSB OF X
                 *
0070 FE 0000   A        LDX     TDATA
0073 A7 00     A        STAA    X         STORE X VALUE
0075 08                 INX               TO DATA BUFF
0076 E7 00     A        STAB    X
0078 08                 INX
0079 FF 0000   A        STX     TDATA     INC BUFF POINTER
                 *
007C B1 0000   A        CMPA    FPCMVU    TEST FOR END PT.
007F 2A 03 0084         BPL     A2        IF=,  TEST LOW BYTE
                 *
0081 7E 0000   A        JMP     RPOL      OTHERWISE RETURN
                 *
0084 F1 0000   A A2     CMPB    FPCMVL    END PT. ?
0087 22 1C 00A5         BHI     CSCAN     YES, END SCAN
                 *
0089 7E 0000   A        JMP     RPOL
                 *
008C 86 04     A OVRFLO LDAA    #$04      END SCAN
008E B7 0000   A        STAA    P2AC
0091 7C 0000   A        INC     SCNFLG    SCAN DONE IF SET
                 *
0094 CE 00BD   P        LDX     #MOVFLO   PRINT OUT OVER FLOW
0097 DF 05     A        STX     $0005     MESSAGE
                 *
0099 BD B2C3   A        JSR     $B2C3
009C BD B2A0   A        JSR     $B2A0
009F BD B38B   A        JSR     $B38B
                 *
00A2 7E 00AD   P        JMP     ENDSC     END SCAN
                 *
00A5 86 04     A CSCAN  LDAA    #$04      SCAN DONE
```

```
 00A7 B7 0000   A         STAA   P2AC     SET INTR. MASK/PIA
                  *
 00AA 7C 0000   A         INC    SCNFLG
                  *
 00AD 7F 0000   A ENDSC   CLR    POLFLG   RESET ENABLE SCAN FLG
 00B0 B6 0000   A         LDAA   P2BP     KEEP 2 MSB OF D/A
 00B3 84 03     A         ANDA   #$03
 00B5 8B 60     A         ADDA   #%01100000 PEN UP,CHART OFF
 00B7 B7 0000   A         STAA   P2BP     ELECT. OFF,SCAN OFF
                  *
 00BA 7E 0000   A         JMP    RPOL
                  *
                          XREF   P1BP,P1AP,P1AC,TDATA,FPCMVU,FPCMVL
                          XREF   P2AC,SCNFLG,RPOL,EPDATA,DIV16
                          XREF   POLFLG,P2BP
                  *
                  *
                          XDEF   NUMBYH,NUMBYL,SCAN
                  *
 00BD     20    A MOVFLO FCC     / **  DATA BUFFER FULL   **
 00BE     2A    A
 00BF     2A    A
 00C0     20    A
 00C1     20    A
 00C2     44    A
 00C3     41    A
 00C4     54    A
 00C5     41    A
 00C6     20    A
 00C7     42    A
 00C8     55    A
 00C9     46    A
 00CA     46    A
 00CB     45    A
 00CC     52    A
 00CD     20    A
 00CE     46    A
 00CF     55    A
 00D0     4C    A
 00D1     4C    A
 00D2     20    A
 00D3     20    A
 00D4     2A    A
 00D5     2A    A
 00D6     20    A
 00D7     20    A
 00D8     20    A
 00D9     20    A
 00DA     20    A
 00DB     20    A
 00DC     20    A
 00DD     20    A
 00DE     20    A
 00DF     20    A
 00E0     20    A
 00E1     20    A
 00E2     20    A
 00E3     20    A
```

AD-A085 989 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 7/2
MICROCOMPUTER-ASSISTED FLOW-THROUGH ASV SYSTEM.(U)
SEP 79 C CLAVELL
UNCLASSIFIED NOSC/TR-532 NL

2 OF 2
AD
A085 989

END
DATE
FILMED
8-80
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963 A

```
 00E4     20     A
 00E5     20     A
                   *
 0000                       DSCT
 0000     0002   A DTEN     RMB     2
 0002     0001   A NUMBYH   RMB     1
 0003     0001   A NUMBYL   RMB     1
                   *
                   *
                            END
ERRORS 00000
```

```
N
 NAM SCAN     VER. 14   5-1Ø-79    CLAVELL
*
*    FILE NAMES:     &SCAN / SCAN (R)
*
 OPT REL
*
 TTL *** DATA COLLECTING ROUTINE ***
*
****************************************
*
* ROUTINE TO COLLECT AND STORE "Y"
* DATA VALUES FOR USE BY COMPUT
* SUBROUTINE.  (1 X-Y VALUE PAIR / .6 SEC)
*
*****************************************
*
*
 PSCT
*
SCAN LDX EPDATA    CHK NEXT TO LAST DATA BUFF
 LDAA X              ADDRESS FOR Ø'S,
 CMPA #Ø             IF NOT Ø - BUFF OVER-
 BEQ OVRF          FLOW
 JMP OVRFLO       GIVE OVER FLOW MESS
*
OVRF LDX #1Ø
 STX DTEN
*
 LDAA #$2Ø   Y AXIS/SET BIT 5
 STAA P1BP
*
 LDAA #$3C    START A/D CONV.
 STAA P1AC
 NOP
 NOP
 LDAA #$34     RESET BIT
 STAA P1AC
CONV LDAA P1AC      CHECK FOR CONV.
 BPL CONV         COMPLETE-BIT 7 SET
*
 LDAA #167
D1 DECA            1 MS DELAY
 BNE D1
*
 LDAA P1BP        GET UPPER BYTE OF Y
 ANDA #$ØF     MASK UPPER 4 BITS
 LDAB P1AP     GET LSB OF Y
*
```

```
************************
*  DIVIDE Y VALUE BY 1Ø
************************
 LDX #DTEN      GET ADDR OF HI
 JSR DIV16         BYTE OF DIVISOR
*
 LDX TDATA    STORE IST Y VALUE
 STAA X         MSB
 INX
 STAB X       LSB
 INX    INCREMENT POINTER
 STX TDATA     SAVE NEW ADDRESS
*
 LDX NUMBYH    INC Y COUNTER
 INX
 STX NUMBYH
*
 CLRA     X AXIS
 STAA P1BP
*
 LDAA #$3C    START A/D CONV.
 STAA P1AC
 NOP
 NOP
 LDAA #$34    RESET BIT
 STAA P1AC
CONV1 LDAA P1AC    CHECK FOR CONV.
 BPL CONV1      COMPLETE-BIT 7 SET
*
 LDAA #167
D2 DECA
 BNE D2
*
 LDAA P1BP  GET MSB OF X
 ANDA #$ØF   MASK UPPER 4 BITS
 LDAB P1AP     GET LSB OF X
*
 LDX TDATA
 STAA X            STORE X VALUE
 INX               TO DATA BUFF
 STAB X
 INX
 STX TDATA           INC BUFF POINTER
*
 JSR OUT1
 JSR OUT
 CMPA FPCMVU    TEST FOR END PT.
 BEQ A2     IF=,  TEST LOW BYTE
*
 JMP RPOL      OTHERWISE RETURN
*
A2 CMPB FPCMVL   END PT. ?
 BCC CSCAN    YES, END SCAN
*
 JMP RPOL
*
CVRFLO LDAA #$Ø4    END SCAN
 STAA P2AC
 INC SCNFLG           SCAN DONE IF SET
*
 LDX #MOVFLO          PRINT OUT OVER FLOW
 STX $ØØØ5            MESSAGE
*
 JSR $B2C3
 JSR $B2AØ
 JSR $B38B
 JMP ENDSC      END SCAN
*
CSCAN LDAA #$04   SCAN DONE
 STAA P2AC      SET INTR. MASK/PIA
 LDAA #$43
 JSR $E1D1  OUT PUT "C"
*
 INC SCNFLG
*
ENDSC CLR POLFLG      RESET ENABLE SCAN FLG
 LDAA P2BP        KEEP 2 MSB OF D/A
 ANDA #$03
 ADDA #%01100000      PEN UP,CHART OFF
 STAA P2BP            ELECT. OFF,SCAN OFF
*
 JMP RPOL
*
OUT STAA SAVA
 STAB SAVB
 LDX #SAVA
 JSR $E0C8
 JSR CR1
 LDAA SAVA
 LDAB SAVB
 RTS
*
OUT1 LDX #FPCMVU
 JSR $E0C8
 JSR CR1
 RTS
*
CR1 LDX #CR
 JSR $E07E
 RTS
*
SAVA RMB 1
SAVB RMB 1
CR FCB $D,$A,4
*
 XREF P1BP,P1AP,P1AC,TDATA,FPCMVU,FPCMVL
 XREF P2AC,SCNFLG,RPOL,EPDATA,DIV16
 XREF POLFLG,P2BP
*
*
 XDEF NUMBYH,NUMBYL,SCAN
*
MOVFLO FCC / **  DATA BUFFER FULL  **
*
 DSCT
DTEN RMB 2
NUMBYH RMB 1
NUMBYL RMB 1
*
*
 END
*
!
```

```
001   HALT   *** PROGRAM HALT ROUTINE ***

                               NAM    HALT     VER.2   8-30-79   CLAVELL
                         *
                         *   FILE NAMES:  &HALT (S) / HALT (R)
                         *
                               OPT    REL
                         *
                               TTL    *** PROGRAM HALT ROUTINE ***
                         *
                         ***************************************************
                         * THIS ROUTINE HALTS THE MAIN PROG AT THE START
                         * OF A NEW CYCLE AND ALLOWS UP TO 20 PARAMETERS
                         * TO BE CHANGED. PROG IS RESTARTED AT CYCLE 1
                         * SET S = NEXT QUESTION TO BE CHNGED
                         * GO = NO MORE CHANGES
                         *        *************************
                         * HALT MAY BE USED TO SIMPLY STOP EXECUTION FOR
                         * A WHILE.  PROG IS RESTARTED WITH "GO".
                         ***************************************************
                         *
                         *
0000                           PSCT
2B34                     *
0000 CE 0000   A HALT      LDX    #MES50    CHANGE MESS.
0003 BD 0000   A           JSR    PRINT
0006 86 01     A           LDAA   #1        1 # BEFORE DEC PT
0008 BD 0000   A           JSR    KBIN      READ KEYBOARD
000B B6 0000   A           LDAA   BINLO
000E 4D                    TSTA             A=1?
000F 26 03 0014            BNE    CHG       YES: CHANGE PARAM
0011 7E 006E   P           JMP    GOCHK     NO: WAIT FOR "GO"
                         *
0014 7F 0000   D CHG       CLR    TX        QUEST COUNTER POINTER
0017 7F 0001   D           CLR    TX+1
001A CE 0000   A CHG1      LDX    #MES79    GET # OF QUEST
001D BD 0000   A           JSR    PRINT     TO BE CHANGED
0020 86 02     A           LDAA   #2
0022 BD 0000   A           JSR    KBIN
                         *
0025 B6 0000   A           LDAA   BINLO
0028 FE 0000   D           LDX    TX
002B A7 00     B           STAA   QBUF,X    QUEST # INTO BUFFER
                         *
002D 7C 0001   D           INC    TX+1      INC POINTER
0030 B6 0001   D           LDAA   TX+1
0033 81 15     A           CMPA   #21       MAX OF 20 PARAM
0035 27 14 004B            BEQ    GONOW     CHANGES
                         *
        70 1D
0037 B6 0000   A           LDAA   P8BP      DUMMY READS FOR
003A B6 0000   A           LDAA   P5BP      SET S & GO
        70 21            *
        70 1F
003D B6 0000   A QLOOP     LDAA   P8BC      "SET S" ?
0040 48                    ASLA
0041 2B D7 001A            BMI    CHG1      YES: NEXT QUEST
                         *
0043 B6 0000   A           LDAA   P5BC      "GO"?
0046 2A F5 003D            BPL    QLOOP     NO: KEEP POLLING
                         *                     YES:NO MORE CHANGES
                         *
```

```
002    HALT    *** PROGRAM HALT ROUTINE ***

 0048 B6 0000  A          LDAA    P5BP      DUMMY READ
                   *
 004B 7A 0001  D GONOW    DEC     TX+1
 004E FE 0000  D          LDX     TX        GET BUFF OFFSET
 0051 A6 00    B          LDAA    QBUF,X
 0053 BD 0000  A          JSR     QFIX      QUEST RETRIEVAL SUBR.
 0056 7D 0001  D          TST     TX+1      DONE?
 0059 26 F0 004B          BNE     GONOW     NO: LOOP BACK
                   *
 005B 7D 0000  A          TST     INTFLG    CALLED FROM INIT?
 005E 26 2D 008D          BNE     GO2       YES: RTS
                   *
 0060 BD 0000  A          JSR     CLSABF    CLR SAMP BUF/COMP
 0063 FE 0000  A          LDX     TIMBUF
 0066 FF 0000  A          STX     TMBUF     REINITIALIZE TIME BUF
 0069 86 01    A          LDAA    #1
 006B B7 0000  A          STAA    CNTR1     RESET CNTR1
                   *
 006E CE 0000  A GOCHK    LDX     #MES60    PUSH "GO" MESSG.
 0071 86 01    A          LDAA    #1
 0073 B7 0000  A          STAA    BLOCK
 0076 BD 0000  A          JSR     SUB3      LF
 0079 BD 0000  A          JSR     PRINT1
 007C BD 0000  A          JSR     SUB3
 007F BD 0000  A          JSR     SUB3
                   *
 0082 7F 0000  A          CLR     HALTF     CLR HALT FLAG
 0085 B6 0000  A          LDAA    P5BP      DUMMY READ
 0088 B6 0000  A GO1      LDAA    P5BC      "GO" PUSHED?
 008B 2A FB 0088          BPL     GO1       NO: LOOP
                   *
 008D 39         GO2      RTS
                   *
                   *
                          XREF    MES50,MES60
                          XREF    PRINT1,PRINT,KBIN,SUB3,P5BP,P5BC
                          XREF    BLOCK,BINLO,BINHI,BCDLO,BCDHI,QFIX
                          XREF    STCN1H,STCN2H,STCN3H,STCN4H,FRSTQF
                          XREF    HALTF,VPRECY,ZINK,TIMBUF,MES79
                          XREF    CLSABF,CNTR1,TMBUF,P8BC,P8BP,INTFLG
                   *
                          XDEF    HALT,CHG,QBUF
                   *
                   *
 0000                     DSCT
                   *
 0000      0002  A TX     RMB     2
                   *
 0000                     BSCT
                   *
 0000      0014  A QBUF   RMB     20        QUEST # BUFF
                   *
                   *
                          END
ERRORS 00000
```

```
001    QFIX    *** QUESTION REPEAT SUBROUTINE ***
                             NAM     QFIX      VER.1    7-30-79   CLAVELL
                      *
                      *   FILE NAMES:   &QFIX (S) / QFIX (R)
                      *
                             OPT     REL
                      *
                             TTL     *** QUESTION REPEAT SUBROUTINE ***
                      *
                      ******************************************************
                      *    ROUTINE TO ENABLE PARAMETER CHANGES
                      *    TO BE MADE. CALLED FROM POLL OR HALT.
                      ******************************************************
                      *
                      *
 0000 7C 0002    D QFIX      INC     F2         SET FLAG
                      *
 0003 48                     ASLA               (A*2)
 0004 B7 0001    D           STAA    CONST+1    OFFSET POINTER
 0007 7F 0000    D           CLR     CONST
 000A FE 0000    D           LDX     CONST      OFFSET INTO X
 000D EE 00      A           LDX     J0,X       ADDR OF QUEST INTO X
                      *                            FROM JUMP TABLE
 000F 6E 00      A           JMP     0,X        JUMP TO QUEST.
                      *
                      *
 0011 7F 0002    D RTNPT     CLR     F2         CLR FLAG
                      *
 0014 39                     RTS
                      *
                      *
                             XREF    J0
                      *
                             XDEF    QFIX,F2,RTNPT
                      *
                      *
 0000                        DSCT
                      *
 0000      0002  A CONST     RMB     2
 0002      0001  A F2        RMB     1
                      *
                             END
ERRORS 00000
```

```
T
 NAM CT   VER.2   4-7-78   CLAVELL
 OPT REL
*
*  FILE NAMES:  &CT (S) / CT (R)
*
 TTL   ***  TOGGLE SUBROUTINE  ***
*
*    TOGGLES SHIFT LINE OF ELAPSED TIMER.
'    USET BY PROGM.
*
CT LDAA #$3E   SET BIT
 STAA P6BC
*
 LDX #23   DELAY
DEL DEX
 BNE DEL
*
 LDAA #$36     CLR BIT
 STAA P6BC
*
 RTS
*
 XREF P6BC
*
 XDEF CT
*
 END
*
!
```

# SECTION IV:
## Computing Routine

```
                        NAM    COMP     VER.3.2    ALLEN   2-7-79
                    *      VER. 19   MODIFIED BY CLAVELL  4-27-79
                        OPT    REL
                    *
                    *  FILE NAMES:  &COMP(S) / COMP(R)
                    *
                        TTL    *** DATA COMPUTING ROUTINE ***
                    *
                    *
00FC                    ORG    $00FC
                    *
00FC    0080   A SABF   RMB    128       PLAIN SAMPLE PEAK AREA
                    *
0000                    BSCT
                    *
0000    0002   A MUL1   RMB    2
0002    0002   A MUL2   RMB    2
                    *           MULTIPLY & DIVIDE BUFF
0004    0002   A MUL3   RMB    2
0006    0002   A MUL4   RMB    2
        0002   B DIV2   EQU    MUL2
        0000   B DIV1   EQU    MUL1
        0004   B DIV3   EQU    MUL3
0008    0010   A CONBF  RMB    16        CONCENTRATION BUFF
0018    0002   A BINBUF RMB    2         BIN TO ASCII CONV.
001A    0008   A LOWECO RMB    8         LOW VALUE FOR EACH ELEMENT
0022    0008   A HIECO  RMB    8         HIGH VALUE FOR EACH ELEMEN
002A    0008   A SAAXA  RMB    8         BUFF FOR DIVISOR
                    *
0000                    DSCT
                    *
0000    0028   A PRBUF  RMB    40
                    *
        0000   A YVAL   EQU    0         //Y VALUE OFFSET IN DATA A
        0002   A XVAL   EQU    2         //X VALUE OFFSET IN DATA A
0A36                *
0028    0004   A AREA   RMB    4
002C    0002   A BINTP1 RMB    2
002E    0002   A BINTP2 RMB    2
0030    0002   A TPC16  RMB    2
0032    0001   A ELNUM1 RMB    1
0033    0001   A ELNUM  RMB    1
0034    0001   A CYCLE  RMB    1
0035    0001   A SUBSMP RMB    1
0036    0001   A SAMPLE RMB    1
0037    0002   A CYNO   RMB    2         CURRENT Y ADDR
0039    0002   A CPTNO  RMB    2         CURRENT X ADDR
003B    0002   A FPTNO  RMB    2         X ADDR OF FIRST POINT ON P
003D    0002   A LPTNO  RMB    2         X ADDR OF LAST POINT ON PE
003F    0002   A CXVAL  RMB    2         CURRENT X VALUE
0041    0002   A FXVAL  RMB    2         FIRST PT IN PEAK
0043    0002   A LXVAL  RMB    2         LAST PT IN PEAK
0045    0001   A CNTR4  RMB    1
0046    0001   A HITCNT RMB    1
0047    0001   A EQ     RMB    1
0048    0002   A PTREG1 RMB    2
004A    0002   A PTREG2 RMB    2
004C    0001   A FLAG1  RMB    1
```

Preceding Page Blank

```
ØØ2   COMP    *** DATA COMPUTING ROUTINE ***

 ØØ4D      ØØØ1  A ONES    RMB     1
 ØØ4E      ØØØ1  A TENS    RMB     1
 ØØ4F      ØØØ1  A HUND    RMB     1
 ØØ5Ø      ØØØ2  A TPT1    RMB     2
 ØØ52      ØØØ2  A TPT2    RMB     2
 ØØ54      ØØØ2  A TPT3    RMB     2
 ØØ56      ØØØ2  A TPT4    RMB     2
           ØØ5Ø  D X1      EQU     TPT1
           ØØ52  D X2      EQU     TPT2
           ØØ54  D Y1      EQU     TPT3
           ØØ56  D Y2      EQU     TPT4
 ØØ58      ØØØ2  A EX      RMB     2
 ØØ5A      ØØØ2  A Y       RMB     2
 ØØ5C      ØØØ1  A SLOPE   RMB     1
 ØØ5D      ØØØ1  A SABOFF  RMB     1
 ØØ5E      ØØØ1  A BITHOU  RMB     1
 ØØ5F      ØØØ1  A BIHUND  RMB     1
 ØØ6Ø      ØØØ1  A BITENS  RMB     1
 ØØ61      ØØØ1  A BIONES  RMB     1
 ØØ62      ØØØ1  A RETF    RMB     1
 ØØ63      ØØØ1  A SIGN    RMB     1            /SIGN FLAG FOR MUL/DIV
 ØØ64      ØØØ4  A XA      RMB     4
 ØØ68      ØØØ2  A SAA     RMB     2
 ØØ6A      ØØØ2  A SAC     RMB     2
 ØØ6C      ØØØ2  A YA      RMB     2
 ØØ6E      ØØØ2  A OR1     RMB     2
 ØØ7Ø      ØØØ2  A R1      RMB     2
 ØØ72      ØØØ2  A R2      RMB     2
 ØØ74      ØØØ1  A ZESUP   RMB     1
 ØØ75      ØØØ1  A ERFLAG  RMB     1
 ØØ76      ØØØ1  A Z1F     RMB     1
 ØØ77      ØØØ1  A CADF    RMB     1
 ØØ78      ØØØ1  A COPF    RMB     1
 ØØ79      ØØØ1  A LEADF   RMB     1
 ØØ7A      ØØ18  A TIMBUF  RMB     24           HR , MIN & SEC BUFF
 ØØ92      ØØØ2  A TMBUF   RMB     2            POINTER TO TIMBUF
 ØØ94      ØØØ2  A SAVX    RMB     2
 ØØ96      ØØØ1  A SAVA    RMB     1
 ØØ97      ØØØ1  A INTCAL  RMB     1
           Ø7C6  A YEND    EQU     $Ø7C6        # OF DATA PT'S AT 1 PT/.6
                   *                          (Ø943 - Ø17D) = Ø7C6 (199Ø)
                   *
 ØØØØ                              PSCT
                   *
                   *********************
                   *ENTRY POINT AND DRIVER
                   **********************
                   *
 ØØØØ 7D ØØØØ  A COMPT     TST     CNTR2        VALID DATA FLAG
                   *                               Ø IF GOOD DATA
 ØØØ3 27 1Ø ØØ15           BEQ     CMPTØ        ITS VALID
 ØØØ5 7F ØØ76  D           CLR     Z1F          CLR ERROR FLAGS
 ØØØ8 7F ØØ77  D           CLR     CADF
 ØØØB 7F ØØ78  D           CLR     COPF
 ØØØE 7F ØØ79  D           CLR     LEADF
                   *
 ØØ11 BD Ø4CF  P           JSR     CLSABF
 ØØ14 39                   RTS                  // NOT VALID
```

```
003   COMP   *** DATA COMPUTING ROUTINE ***

                        ********************
                        *FIND FIRST PEAK IN CURVE
                        ********************
                        *
 0015 CE 0000  A CMPT0  LDX    #DATA    /FIRST POINT TO LOOK AT
 0018 FF 0037  D        STX    CYNO
 001B 7F 0063  D        CLR    SIGN
 001E BD 0093  P CMPT1  JSR    FINDPK   /FIND FIRST PEAK IN CURVE
 0021 7D 0033  D        TST    ELNUM    /THIS PEAK # 0=NONE
 0024 27 0B 0031        BEQ    CMPT2    /NO MORE
 0026 BD 0349  P        JSR    ADJUST   /ADJUST LINE AT BOTTOM
 0029 BD 0400  P        JSR    INTGRT   /INTEGRATE ONE PEAK
 002C BD 04DA  P        JSR    SAVRES   /SAVE INTEGRATED RESULTS
 002F 20 ED 001E        BRA    CMPT1    /NEXT PEAK
 0031 B6 0000  A CMPT2  LDAA   CNTR1
 0034 B1 0000  A        CMPA   VPRECY   /NEED UP TO 8 CYCLES
 0037 27 19 0052        BEQ    CMPT3    /READY TO CALC CONCS
                        *
 0039 FE 0092  D        LDX    TMBUF    SAVE SAMPLE AQUIS. TIME
 003C B6 0000  A        LDAA   HR
 003F A7 00    A        STAA   0,X
 0041 08                INX
 0042 B6 0000  A        LDAA   MIN
 0045 A7 00    A        STAA   0,X
 0047 08                INX
 0048 B6 0000  A        LDAA   SEC
 004B A7 00    A        STAA   0,X
 004D 08                INX
 004E FF 0092  D        STX    TMBUF    UPDATE POINTER
                        *
 0051 39                RTS             /NEED /MORE CYCLES
                        ********************
                        *CALCULATE CONCENTRATIONS
                        ********************
 0052 7F 0035  D CMPT3  CLR    SUBSMP
 0055 7F 0097  D        CLR    INTCAL
 0058 7F 0074  D        CLR    ZESUP
 005B CE 007A  D        LDX    #TIMBUF  INITIALIZE POINTER
 005E FF 0092  D        STX    TMBUF
 0061 7C 0097  D        INC    INTCAL
 0064 BD 055B  P        JSR    CLSXA    CLR DIVISOR BUFF
 0067 BD 0525  P CMPT4  JSR    CALCON   /CALCULATE CONCENTRATIONS
 006A 7D 0097  D        TST    INTCAL
 006D 27 08 0077        BEQ    CMPT6
 006F 7F 0097  D        CLR    INTCAL
 0072 7F 0035  D        CLR    SUBSMP
 0075 20 F0 0067        BRA    CMPT4
                        *
 0077 BD 070D  P CMPT6  JSR    PRTSMP   /PRINT SAMPLE
 007A 7C 0036  D CMPT5  INC    SAMPLE
 007D 7C 0035  D        INC    SUBSMP
 0080 B6 0000  A        LDAA   VPRECY   //LAST SAMPLE
 0083 4A                DECA
 0084 B1 0035  D        CMPA   SUBSMP
 0087 26 DE 0067        BNE    CMPT4    /NO, NEXT ONE
 0089 BD 04CF  P        JSR    CLSABF
 008C CE 007A  D        LDX    #TIMBUF
 008F FF 0092  D        STX    TMBUF    REINITIALIZE POINTER
```

```
0092 39                RTS              /YES, DONE WITH THIS GROUP
                ********************
                *FIND THE NEXT PEAK IN THE CURVE
                *PUT
                *FIRST POINT OFFSET IN FPTNO
                *LAST POINT OFFSET IN LPTNO
                *ELEMENT # IN ELNUM
                *SET ELEMENT # TO 0 IF NO MORE PEAKS
                ********************
                *
                *
                *
0093 BD 01CF  P FINDPK JSR    FNDPOS   //FIND A POSITIVE SLOPE
0096 5D                TSTB
0097 27 2B 00C4        BEQ    FIND3    //END OF CURVES
0099 FE 0037  D        LDX    CYNO     /REMEMBER FIRST POINT
009C 08                INX
009D 08                INX
009E FF 003B  D        STX    FPTNO    /IN THIS PEAK
00A1 FE 003F  D        LDX    CXVAL
00A4 FF 0041  D        STX    FXVAL
00A7 BD 00F8  P        JSR    FNDEND   FIND END PT OF CURVE
00AA 5D                TSTB
00AB 27 17 00C4        BEQ    FIND3
00AD FE 0037  D        LDX    CYNO     /LAST POINT IN THIS PEAK
00B0 08                INX
00B1 08                INX
00B2 FF 003D  D        STX    LPTNO
00B5 FE 003F  D        LDX    CXVAL    /LAST X AXIS VALUE
00B8 FF 0043  D        STX    LXVAL
                *
00BB BD 02EF  P        JSR    WCHELM   DECIDE WHICH ELEMENT
                *
00BE 7D 0033  D        TST    ELNUM
00C1 27 05 00C8        BEQ    FIND2
                *
00C3 39                RTS
00C4 7F 0033  D FIND3  CLR    ELNUM    //NO MORE CURVES
00C7 39                RTS
                * CHECK FOR END OF DATA
00C8 BD 00D1  P FIND2  JSR    ENDAT
00CB 5D                TSTB
00CC 27 F6 00C4        BEQ    FIND3    //OUT OF POINTS -- RETURN
00CE 7E 0093  P        JMP    FINDPK   //KEEP LOOKING
                *
                * CHECK FOR END OF DATA
                *  NEED AT LEAST 10 MORE VALID Y POINTS
                *  NEED CPTNO <= YEND
                * RETURN:
                *   0 IF AT END OF DATA
                *   1 IF NOT AT END OF DATA
                *
                *
                *
00D1 CE 0000  A ENDAT  LDX    #DATA
00D4 FF 0000  A        STX    TDATA    CURRENT PT ADDR.
                *                         MINUS STARTING PT ADDR
00D7 B6 0037  D        LDAA   CYNO     = # OF PT'S CH'ED SO FAR
```

```
ØØ5    COMP    *** DATA COMPUTING ROUTINE ***

 ØØDA F6 ØØ38   D           LDAB    CYNO+1
                      *
 ØØDD FØ ØØØ1   A           SUBB    TDATA+1
 ØØEØ B2 ØØØØ   A           SBCA    TDATA
                      *
 ØØE3 CB 28     A           ADDB    #4Ø
 ØØE5 24 Ø1 ØØE8            BCC     ENDAT1     //CARRY
 ØØE7 4C                    INCA
 ØØE8 CE Ø7C6   A  ENDAT1   LDX     #YEND      //NUMBER OF ADDR.'S TO END
 ØØEB BD ØAA9   P           JSR     CMP16      //COMPARE THEM
 ØØEE C1 FF     A           CMPB    #$FF
 ØØFØ 26 Ø3 ØØF5            BNE     ENDAT2     //NOT END YET
 ØØF2 C6 ØØ     A           LDAB    #Ø
 ØØF4 39                    RTS
 ØØF5 C6 Ø1     A  ENDAT2   LDAB    #1
 ØØF7 39                    RTS
                      *
                      *
                      * MOVE ALONG THE PEAK FOUND BY FNDPOS
                      * UNTIL FIND THE END OF PEAK THEN STORE
                      * THAT POINT IN CYNO & CXVAL
                      *
                      *
 ØØF8 7F ØØ4C   D  FNDEND   CLR     FLAG1
 ØØFB 7F ØØ45   D  C1Ø      CLR     CNTR4
 ØØFE 7F ØØ46   D           CLR     HITCNT
 Ø1Ø1 7F ØØ47   D           CLR     EQ
                      *
 Ø1Ø4 FE ØØ37   D  C15      LDX     CYNO
 Ø1Ø7 FF ØØ48   D           STX     PTREG1
 Ø1ØA Ø8                    INX                SET UP TEMP Y VALUE
 Ø1ØB Ø8                    INX                BUFF
 Ø1ØC Ø8                    INX
 Ø1ØD Ø8                    INX
 Ø1ØE FF ØØ4A   D           STX     PTREG2
                      *
 Ø111 7C ØØ45   D  C2Ø      INC     CNTR4      START LOOP
                      *
 Ø114 BD Ø19E   P           JSR     NGPTCP
                      *
 Ø117 81 Ø2     A           CMPA    #2         PT1>PT2 ?
 Ø119 27 17 Ø132            BEQ     HIT3
 Ø11B 7D ØØ4C   D           TST     FLAG1      NO: DO WE HAVE 2 HITS YET?
 Ø11E 26 Ø4 Ø124            BNE     SKA1
 Ø12Ø 81 Ø1     A           CMPA    #1         NO: PT1=PT2 ?
 Ø122 26 3C Ø16Ø            BNE     C25
                      *
 Ø124 B6 ØØ47   D  SKA1     LDAA    EQ         YES:
 Ø127 81 Ø5     A           CMPA    #5         DO WE HAVE 5 PT'S IN LINE?
 Ø129 27 3E Ø169            BEQ     SUC        YES:  SUCCESS,FOUND END
                      *
 Ø12B 7D ØØ4C   D           TST     FLAG1      2 HITS YET?
 Ø12E 26 14 Ø144            BNE     A1Ø
 Ø13Ø 2Ø ØB Ø13D            BRA     C3Ø
                      *
 Ø132 7C ØØ46   D  HIT3     INC     HITCNT
 Ø135 7F ØØ47   D           CLR     EQ
 Ø138 7D ØØ4C   D           TST     FLAG1
```

```
013B 26 07 0144           BNE    A10
                  *
013D B6 0046    D C30     LDAA   HITCNT    2 HITS YET?
0140 81 02      A         CMPA   #2        YES: THIS IS TOP OF PEAK
0142 27 48 018C           BEQ    C40
                  *
0144 BD 02AD    P A10     JSR    INCREG    MOVE TO NEXT PT
0147 B6 0045    D         LDAA   CNTR4
014A 81 0A      A         CMPA   #10       DONE 10 LOOPS YET?
014C 26 C3 0111           BNE    C20       NO: LOOP AGAIN
                  *
014E B6 0046    D         LDAA   HITCNT    YES: 5 OR MORE HITS?
0151 81 04      A         CMPA   #4
0153 2E 22 0177           BGT    C35
                  *
0155 FE 0037    D         LDX    CYNO      NO: UPDATE X VALUE
0158 EE 02      A         LDX    XVAL,X
015A FF 003F    D         STX    CXVAL
                  *
015D C6 01      A         LDAB   #1        SUCCESS - RETURN
015F 39                   RTS
                  *
0160 BD 02CC    P C25     JSR    NEXTPT    MOVE TO NEXT PT
0163 5D                   TSTB             END OF DATA BUFF ?
0164 27 25 018B           BEQ    C45       YES: DONE- RETURN
0166 7E 00F8    P         JMP    FNDEND    START OVER
                  *
0169 FE 004A    D SUC     LDX    PTREG2    FOUND ENDPT OF PEAK
016C FF 0037    D         STX    CYNO
016F EE 02      A         LDX    XVAL,X
0171 FF 003F    D         STX    CXVAL
                  *
0174 C6 01      A         LDAB   #1
0176 39                   RTS
                  *
0177 BD 00D1    P C35     JSR    ENDAT     END OF DATA BUFF?
                  *
017A 5D                   TSTB
017B 27 0E 018B           BEQ    C45
017D FE 004A    D         LDX    PTREG2    NO: RESET CYNO TO
0180 FF 0037    D         STX    CYNO      LAST PT
0183 EE 02      A         LDX    XVAL,X
0185 FF 003F    D         STX    CXVAL     UPDATE X VALUE
                  *
0188 7E 00FB    P         JMP    C10       START OVER
                  *
018B 39           C45     RTS
                  *
018C FE 004A    D C40     LDX    PTREG2    RESET CYNO
018F FF 0037    D         STX    CYNO
0192 7C 004C    D         INC    FLAG1
0195 7F 0045    D         CLR    CNTR4
0198 7F 0047    D         CLR    EQ
019B 7E 0104    P         JMP    C15
                  *
                  *
019E FE 0048    D NGPTCP  LDX    PTREG1
01A1 A6 00      A         LDAA   0,X
```

```
007   COMP    *** DATA COMPUTING ROUTINE ***

 01A3 E6 01      A         LDAB   1,X
 01A5 FE 004A    D         LDX    PTREG2
 01A8 EE 00      A         LDX    0,X
                   *
 01AA BD 0AA9    P         JSR    CMP16
                   *
 01AD 5D                   TSTB
 01AE 27 09 01B9           BEQ    C55       X=AB ?
 01B0 C1 01      A         CMPB   #1        X>AB
 01B2 27 03 01B7           BEQ    C50
 01B4 86 02      A         LDAA   #2        X<AB
                   *
 01B6 39                   RTS
                   *
 01B7 4F           C50     CLRA
 01B8 39                   RTS
                   *
 01B9 7C 0047    D C55     INC    EQ
 01BC 86 01      A         LDAA   #1
 01BE 39                   RTS
                   *
                   *
 01BF FE 0039    D NEXPNT  LDX    CPTNO
 01C2 08                   INX
 01C3 08                   INX
 01C4 08                   INX
 01C5 08                   INX
 01C6 FF 0039    D         STX    CPTNO
                   *
 01C9 EE 00      A         LDX    0,X
 01CB FF 003F    D         STX    CXVAL
                   *
 01CE 39                   RTS
                   *
                   *
                   *
                   *
                   * SEARCH FOR THE BEGINNING OF A PEAK DEFINED
                   *  BY 5  POINTS WITH POSITIVE SLOPE OUT OF 10
                   *  CONSECUTIVE POINTS STARTING AT CURRENT POINT.
                   *
 01CF 7F 0045    D FNDPOS  CLR    CNTR4
 01D2 7F 0046    D         CLR    HITCNT
 01D5 7F 0047    D         CLR    EQ
                   *
 01D8 FE 0037    D         LDX    CYNO
 01DB FF 0048    D         STX    PTREG1    INITIALIZE PTREG
 01DE 08                   INX
 01DF 08                   INX              & PTREG2 TO 1ST
 01E0 08                   INX              & 2ND PT'S TO
 01E1 08                   INX              COMPARE
 01E2 FF 004A    D         STX    PTREG2
                   *
 01E5 7C 0045    D ENT     INC    CNTR4
                   *
 01E8 BD 0234    P         JSR    PTCMP     DO PT COMPARISONS
                   *
 01EB 81 02      A         CMPA   #2        A=2?
```

```
01ED 27 0D 01FC          BEQ    HIT       YES: INC HIT COUNTER
01EF 81 01      A        CMPA   #1        A=1?
01F1 26 17 020A          BNE    FP        NO: CHK FOR BUFF END
                  *
01F3 B6 0047    D        LDAA   EQ        EQ = 3?
01F6 81 03      A        CMPA   #3
01F8 27 0D 0207          BEQ    UPD       YES: UPDATE FIRST PT & STA
01FA 20 06 0202          BRA    ICREG
                  *
01FC 7C 0046    D HIT    INC    HITCNT
01FF 7F 0047    D        CLR    EQ
                  *
0202 BD 02AD    P ICREG  JSR    INCREG
0205 20 0D 0214          BRA    HIT1
                  *
0207 BD 02A1    P UPD    JSR    LNUPDT    MOVE TO NEXT PT
                  *
020A BD 00D1    P FP     JSR    ENDAT     CHK FOR END OF DATA BUFF
020D 5D                  TSTB
020E 26 01 0211          BNE    F1        NOT DONE YET
0210 39                  RTS              EMPTY-RETURN
0211 7E 01CF    P F1     JMP    FNDPOS
                  *
                  *                      ONE PT OVER
0214 B6 0046    D HIT1   LDAA   HITCNT    DO WE HAVE 5 PT'S?
0217 81 05      A        CMPA   #5        WITH POS SLOPE?
0219 26 0B 0226          BNE    LOOP1     NO: MAKE ANOTHER LOOP
                  *
021B C6 FF      A        LDAB   #$FF      YES: SUCCESS
021D FE 0037    D        LDX    CYNO
0220 EE 02      A        LDX    XVAL,X    UPDATE X VALUE
0222 FF 003F    D        STX    CXVAL
                  *
0225 39                  RTS
                  *
0226 B6 0045    D LOOP1  LDAA   CNTR4     HAVE WE CHK'ED 10 PT'S?
0229 81 0A      A        CMPA   #10
022B 27 03 0230          BEQ    NP1       YES START AGAIN:
                  *
022D 7E 01E5    P        JMP    ENT
                  *
0230 BD 02C2    P NP1    JSR    FNDP2     YES: MOVE OVER 1 PT
                  *                     AND REPEAT
0233 39                  RTS
                  *
                  *
0234 FE 004A    D PTCMP  LDX    PTREG2
0237 A6 00      A        LDAA   0,X
0239 E6 01      A        LDAB   1,X
023B FE 0048    D        LDX    PTREG1
023E EE 00      A        LDX    0,X
                  *
0240 BD 0AA9    P        JSR    CMP16
                  *
0243 5D                  TSTB
0244 27 0B 0251          BEQ    EQINC     X=AB
0246 C1 01      A        CMPB   #1        X>AB
0248 27 03 024D          BEQ    CK10
```

```
ØØ9   COMP    *** DATA COMPUTING ROUTINE ***

 Ø24A 86 Ø2      A         LDAA    #2          X<AB
 Ø24C 39                   RTS
                 *
 Ø24D BD Ø257    P CK1Ø    JSR     CKNXPT
 Ø25Ø 39                   RTS
                 *
 Ø251 7C ØØ47    D EQINC   INC     EQ
 Ø254 86 Ø1      A         LDAA    #1
 Ø256 39                   RTS
                 *
                 *
 Ø257 FE ØØ4A    D CKNXPT  LDX     PTREG2
 Ø25A Ø8                   INX                 GET NEXT Y VALUE ADR
 Ø25B Ø8                   INX
 Ø25C Ø8                   INX
 Ø25D Ø8                   INX
 Ø25E FF ØØ4A    D         STX     PTREG2
                 *
 Ø261 FE ØØ4A    D         LDX     PTREG2      COMPARE 1ST & 3RD PT'S
 Ø264 A6 ØØ      A         LDAA    Ø,X
 Ø266 E6 Ø1      A         LDAB    1,X
 Ø268 FE ØØ48    D         LDX     PTREG1
 Ø26B EE ØØ      A         LDX     Ø,X
                 *
 Ø26D BD ØAA9    P         JSR     CMP16
                 *
 Ø27Ø 5D                   TSTB
 Ø271 27 1E 0291           BEQ     EQINC1      X=AB -  RETURN 1 IN A
 Ø273 C1 FF      A         CMPB    #$FF        X<AB - RETURN 2 IN A
 Ø275 27 ØD 0284           BEQ     HIT2
                 *                             X>AB -  RETURN Ø IN A
 Ø277 FE ØØ4A    D         LDX     PTREG2      RESET CYNO
 Ø27A FF ØØ37    D         STX     CYNO
 Ø27D EE Ø2      A         LDX     XVAL,X
 Ø27F FF ØØ3F    D         STX     CXVAL
 Ø282 4F                   CLRA
 Ø283 39                   RTS
                 *
 Ø284 FE ØØ48    D HIT2    LDX     PTREG1
 Ø287 Ø8                   INX
 Ø288 Ø8                   INX
 Ø289 Ø8                   INX
 Ø28A Ø8                   INX
 Ø28B FF ØØ48    D         STX     PTREG1
                 *
 Ø28E 86 Ø2      A         LDAA    #2
 Ø29Ø 39                   RTS
                 *
 Ø291 7C ØØ47    D EQINC1  INC     EQ
 Ø294 FE ØØ48    D         LDX     PTREG1
 Ø297 Ø8                   INX
 Ø298 Ø8                   INX
 Ø299 Ø8                   INX
 Ø29A Ø8                   INX
 Ø29B FF ØØ48    D         STX     PTREG1
 Ø29E 86 Ø1      A         LDAA    #1
 Ø2AØ 39                   RTS
                 *
```

```
                     *
 Ø2A1 FE ØØ48   D LNUPDT LDX    PTREG1
 Ø2A4 FF ØØ37   D        STX    CYNO     RESET CYNO
                     *
 Ø2A7 EE Ø2     A        LDX    XVAL,X
 Ø2A9 FF ØØ3F   D        STX    CXVAL    UPDATE X VALUE
                     *
 Ø2AC 39                 RTS
                     *
                     *
 Ø2AD FE ØØ48   D INCREG LDX    PTREG1
 Ø2BØ Ø8                 INX
 Ø2B1 Ø8                 INX
 Ø2B2 Ø8                 INX
 Ø2B3 Ø8                 INX
 Ø2B4 FF ØØ48   D        STX    PTREG1
                     *
 Ø2B7 FE ØØ4A   D        LDX    PTREG2
 Ø2BA Ø8                 INX
 Ø2BB Ø8                 INX
 Ø2BC Ø8                 INX
 Ø2BD Ø8                 INX
 Ø2BE FF ØØ4A   D        STX    PTREG2
                     *
 Ø2C1 39                 RTS
                     *
                     *
 Ø2C2 BD Ø2CC   P FNDP2  JSR    NEXTPT
 Ø2C5 5D                 TSTB
 Ø2C6 26 Ø1 Ø2C9         BNE    ME2
 Ø2C8 39                 RTS
 Ø2C9 7E Ø1CF   P ME2    JMP    FNDPOS
                     *
                     *GO TO NEXT POINT (TO RIGHT ALONG X AXIS)
 Ø2CC FE ØØ37   D NEXTPT LDX    CYNO     /CURRENT POINT #
 Ø2CF Ø8                 INX
 Ø2DØ Ø8                 INX
 Ø2D1 Ø8                 INX
 Ø2D2 Ø8                 INX
 Ø2D3 FF ØØ37   D        STX    CYNO     /NEXT POINT
 Ø2D6 EE Ø2     A        LDX    XVAL,X
 Ø2D8 FF ØØ3F   D        STX    CXVAL
 Ø2DB BD ØØD1   P        JSR    ENDAT    //CHECK FOR END OF YO1
 Ø2DE 39                 RTS
                     ********************
                     *GO TO PREVIOUS POINT (TO LEFT ALONG X AXIS)
                     ********************
 Ø2DF FE ØØ39   D LASTPT LDX    CPTNO
 Ø2E2 Ø9                 DEX
 Ø2E3 Ø9                 DEX
 Ø2E4 Ø9                 DEX
 Ø2E5 Ø9                 DEX
 Ø2E6 FF ØØ39   D        STX    CPTNO
 Ø2E9 EE ØØ     A        LDX    Ø,X
 Ø2EB FF ØØ3F   D        STX    CXVAL
 Ø2EE 39                 RTS
                     ********************
                     *DECIDE WHICH ELEMENT THE CURRENT
```

```
                         *PEAK REPRESENTS IF ANY
                         *RETURN:
                         * Ø - NOT A VALID PEAK
                         * 2 - ZINC Ø
                         * 4 - CADMIUM 2
                         * 6 - LEAD 4
                         * 8 - COPPER 6
                         ********************
 Ø2EF 86 Ø8       A WCHELM LDAA   #8
 Ø2F1 B7 ØØ33   D          STAA   ELNUM      /TRY ALL 4 ELEMENTS
 Ø2F4 CE ØØØ6   A          LDX    #6
 Ø2F7 FF ØØ56   D WCHEØ    STX    TPT4       /SAVE ELEMENT INDEX
 Ø2FA EE 1A     B          LDX    LOWECO,X   /LOW POSSIBLE
 Ø2FC B6 ØØ41   D          LDAA   FXVAL      /FISRT POINT IN PEAK
 Ø2FF F6 ØØ42   D          LDAB   FXVAL+1
 Ø3Ø2 BD ØAA9   P          JSR    CMP16
 Ø3Ø5 C1 Ø1     A          CMPB   #1         IF =1 NO FIT
 Ø3Ø7 27 19 Ø322           BEQ    WCHE1      MUST BE (X<=AB)
 Ø3Ø9 FE ØØ56   D          LDX    TPT4
 Ø3ØC EE 22     B          LDX    HIECO,X    /HIGH POSSIBLE
 Ø3ØE B6 ØØ43   D          LDAA   LXVAL
 Ø311 F6 ØØ44   D          LDAB   LXVAL+1
                         *
 Ø314 4D                   TSTA
 Ø315 26 Ø3 Ø31A           BNE    WCHE6      STILL HAVE DATA
 Ø317 5D                   TSTB
 Ø318 27 2B Ø345           BEQ    WCHE5      NO MORE DATA: STOP
                         *
 Ø31A BD ØAA9   P WCHE6    JSR    CMP16      MUST BE (X> OR = AB)
 Ø31D C1 FF     A          CMPB   #$FF
 Ø31F 27 Ø1 Ø322           BEQ    WCHE1
 Ø321 39          WCHE2    RTS
 Ø322 7D ØØØØ   A WCHE1    TST    ZINK
 Ø325 27 ØA Ø331           BEQ    WCHE3
                         *
 Ø327 7A ØØ33   D          DEC    ELNUM
 Ø32A 7A ØØ33   D          DEC    ELNUM
 Ø32D 27 F2 Ø321           BEQ    WCHE2      /NOT A VALID PEAK
 Ø32F 2Ø ØD Ø33E           BRA    WCHE4
                         *
 Ø331 7A ØØ33   D WCHE3    DEC    ELNUM      IF NOT DOING ZN
 Ø334 7A ØØ33   D          DEC    ELNUM      DON'T CMP. RANGE
 Ø337 B6 ØØ33   D          LDAA   ELNUM
 Ø33A 81 Ø2     A          CMPA   #2
 Ø33C 27 Ø7 Ø345           BEQ    WCHE5
                         *
 Ø33E FE ØØ56   D WCHE4    LDX    TPT4
 Ø341 Ø9                   DEX
 Ø342 Ø9                   DEX
 Ø343 2Ø B2 Ø2F7           BRA    WCHEØ
 Ø345 7F ØØ33   D WCHE5    CLR    ELNUM
 Ø348 39                   RTS
                         *
                         ********************
                         *ADJUST THE LINE AT THE BOTTOM OF CURVE
                         *TO INSURE THAT NO POINTS ON THE
                         *CURVE ARE BELOW THE LINE
                         *ASSUMES:
```

```
                      * FPTNO CONTAINS THE FIRST POINT ADDR
                      * FXVAL CONTAINS THE FIRST X VALUE
                      * LPTNO CONTAINS THE LAST POINT ADDR
                      * LXVAL CONTAINS THE LAST X VALUE
                      *RETURNS:
                      * FPTNO,FXVAL,LPTNO,LXVAL DEFINING
                      *THE END POINTS OF THE LINE AT THE BOTTOM
                      *OF THE CURVE WITH NO POINTS MISSING.
                      ************************************
 0349 FE 003B  D ADJUST LDX    FPTNO     /LEFT END PT OF LINE
 034C FF 0039  D        STX    CPTNO
 034F FE 0041  D        LDX    FXVAL     /X VALUE AT LEFT END OF LI
 0352 FF 003F  D        STX    CXVAL     /
 0355 BD 01BF  P        JSR    NEXPNT    /LOOK AT NEXT PT TO RIGHT
 0358 BD 0390  P        JSR    ABLINE    /IS THIS PT ABOVE THE LINE
 035B 4D                TSTA             /#1=TRUE
 035C 26 0E 036C        BNE    ADJ5      /ADJUST RIGHT END
 035E FE 0039  D        LDX    CPTNO     /NEW LEFT END
 0361 FF 003B  D        STX    FPTNO
 0364 FE 003F  D        LDX    CXVAL
 0367 FF 0041  D        STX    FXVAL
 036A 20 DD 0349        BRA    ADJUST    /TRY AGAIN
 036C FE 003D  D ADJ5   LDX    LPTNO     /RIGHT END PT
 036F FF 0039  D        STX    CPTNO
 0372 FE 0043  D        LDX    LXVAL
 0375 FF 003F  D        STX    CXVAL
 0378 BD 02DF  P        JSR    LASTPT    /MOVE ONE PT TO LEFT
 037B BD 0390  P        JSR    ABLINE    /IS THIS PT ABOVE THE LINE
 037E 4D                TSTA
 037F 26 0E 038F        BNE    ADJ6
 0381 FE 0039  D        LDX    CPTNO
 0384 FF 003D  D        STX    LPTNO
 0387 FE 003F  D        LDX    CXVAL
 038A FF 0043  D        STX    LXVAL
 038D 20 DD 036C        BRA    ADJ5
 038F 39         ADJ6   RTS
                      ********************
                      *DETERMINES IF THE POINT DEFINED BY
                      *CPTNO AND CXVAL IS ABOVE THE LINE
                      *DETERMINED BY FPTNO,FXVAL AND LPTNO,LXVAL
                      *RETURN 1=TRUE 0=FALSE IN ACCA
                      *Y VALUE ON LINE IS LEFT IN Y
                      ********************
 0390 FE 003B  D ABLINE LDX    FPTNO     /GET Y1
 0393 09                DEX              1ST Y VALUE
 0394 09                DEX
 0395 EE 00    A        LDX    0,X
 0397 FF 0054  D        STX    Y1
 039A FE 003D  D        LDX    LPTNO     /GET Y2
 039D 09                DEX              LAST Y VALUE
 039E 09                DEX
 039F EE 00    A        LDX    0,X
 03A1 FF 0056  D        STX    Y2
                      *
 03A4 B6 0057  D        LDAA   Y2+1      /FORM (Y2-Y1)
 03A7 B0 0055  D        SUBA   Y1+1
 03AA 97 01    B        STAA   MUL1+1
 03AC B6 0056  D        LDAA   Y2
```

```
03AF B2 ØØ54  D         SBCA   Y1
03B2 97 ØØ    B         STAA   MUL1
03B4 B6 ØØ4Ø  D         LDAA   CXVAL+1    /FORM (XC-X1)
03B7 BØ ØØ42  D         SUBA   FXVAL+1
03BA 97 Ø5    B         STAA   MUL3+1
03BC B6 ØØ3F  D         LDAA   CXVAL
03BF B2 ØØ41  D         SBCA   FXVAL
03C2 97 Ø4    B         STAA   MUL3
                  *
03C4 BD Ø997  P         JSR    MUL        /MULTIPLY (Y2-Y1)(XC-X1)
                  *
03C7 B6 ØØ44  D         LDAA   LXVAL+1    /FORMS (X2-X1)
03CA BØ ØØ42  D         SUBA   FXVAL+1
03CD F6 ØØ43  D         LDAB   LXVAL      /DIVISOR IN A(LO) &
03DØ F2 ØØ41  D         SBCB   FXVAL      / B (HI)
                  *
03D3 BD Ø9F1  P         JSR    DDIV       /DIVIDE(Y2-Y1)(X-X1) BY (X
                  *
03D6 96 Ø5    B         LDAA   DIV3+1     /ADD Y1 TO RESULT
03D8 BB ØØ55  D         ADDA   Y1+1
03DB B7 ØØ5B  D         STAA   Y+1        (SLOPE * X)
03DE 96 Ø4    B         LDAA   DIV3
03EØ B9 ØØ54  D         ADCA   Y1
03E3 B7 ØØ5A  D         STAA   Y          STORE IN 'Y'
03E6 FE ØØ39  D         LDX    CPTNO      /CURRENT PT #
03E9 Ø9                 DEX               GET Y VALUE AT THIS PT
03EA Ø9                 DEX
03EB EE ØØ    A         LDX    Ø,X
03ED B6 ØØ5A  D         LDAA   Y
03FØ F6 ØØ5B  D         LDAB   Y+1
03F3 BD ØAA9  P         JSR    CMP16      /COMPARE TO CALCULATED VAL
03F6 C1 FF    A         CMPB   #$FF
03F8 26 Ø3 Ø3FD         BNE    ABL1       /POINT IS ABOVE OR EQUAL T
03FA 86 ØØ    A         LDAA   #Ø         /RETURN FALSE IF PT IS < L
03FC 39                 RTS
03FD 86 Ø1    A ABL1    LDAA   #1         /RETURN TRUE IF PT > OR =
03FF 39                 RTS
                  *******************
                  *CALCULATE THE AREA UNDER THE FUNCTION CURVE
                  *DELINEATED BY FPTNO,FXVAL AND LPTNO,LXVAL.
                  *THIS ROUTINE USES THE TRAPEZOIDAL RULE FOR
                  *STEP-WISE INTEGRATION APPROXIMATION.
                  *******************
Ø4ØØ FE ØØ3B  D INTGRT  LDX    FPTNO      /START OF POSITION IN CURV
Ø4Ø3 FF ØØ39  D         STX    CPTNO
Ø4Ø6 FE ØØ41  D         LDX    FXVAL
Ø4Ø9 FF ØØ3F  D         STX    CXVAL
Ø4ØC 7F ØØ5C  D         CLR    SLOPE
Ø4ØF FE ØØ3D  D         LDX    LPTNO      DETERMINE BASE LINE
Ø412 Ø9                 DEX               SLOPE SIGN
Ø413 Ø9                 DEX
Ø414 A6 ØØ    A         LDAA   Ø,X
Ø416 E6 Ø1    A         LDAB   1,X        LAST Y VALUE
                  *
Ø418 FE ØØ3B  D         LDX    FPTNO
Ø41B Ø9                 DEX
Ø41C Ø9                 DEX
Ø41D EE ØØ    A         LDX    Ø,X        1ST Y VALUE
```

```
                        *
 041F BD 0AA9  P               JSR    CMP16
 0422 C1 01    A               CMPB   #1
 0424 26 03 0429               BNE    INTG1      /SLOPE IS POSITIVE
 0426 7C 005C  D               INC    SLOPE      /SLOPE IS NEGATIVE
 0429 CE 0000  A INTG1         LDX    #0
 042C FF 0028  D               STX    AREA       /CLEAR AREA BUFFER
 042F FF 002A  D               STX    AREA+2
 0432 BD 0446  P INTG2         JSR    GTAREA     /CALC AREA OF ONE TRAPEZOI
 0435 FE 0039  D               LDX    CPTNO
 0438 B6 003D  D               LDAA   LPTNO
 043B F6 003E  D               LDAB   LPTNO+1
 043E BD 0AA9  P               JSR    CMP16      /ARE WE THRU?
 0441 C1 FF    A               CMPB   #$FF
 0443 27 ED 0432               BEQ    INTG2      /NO, DO NEXT TRAPEZOID
 0445 39                       RTS               /THRU WITH THIS CURVE
                 ********************
                 *THIS ROUTINE CALCULATES THE AREA OF ONE
                 *TRAPEZOID DEFINED BY CPTNO AND CPTNO+1 AND
                 *ADDS THIS CALCULATED AREA TO AREA
                 **THE TRAPEZOIDAL RULE FOR STEP-WISE INTEGRATION
                 ********************
 0446 7D 005C  D GTAREA TST    SLOPE      /IS SLOPE OF LINE POSITIVE
 0449 27 03 044E               BEQ    GTAR1      /YES
 044B BD 01BF  P               JSR    NEXPNT     /NEGATIVE - USE RIGHT SIDE
 044E BD 0390  P GTAR1         JSR    ABLINE     /CALC Y VALUE ON BASE LINE
 0451 7D 005C  D               TST    SLOPE
 0454 27 03 0459               BEQ    GTAR2
 0456 BD 02DF  P               JSR    LASTPT     /BACK TO LEFT SIDE
 0459 FE 0039  D GTAR2         LDX    CPTNO      / CURRENT X ADDR
 045C 09                       DEX               Y-ADDR.
 045D 09                       DEX
 045E A6 01    A               LDAA   1,X        /LOW  BYTE (Y VALUE)
 0460 B0 005B  D               SUBA   Y+1        /SUBTRACT BASE LINE
 0463 B7 0055  D               STAA   Y1+1       /Y VALUE
 0466 A6 00    A               LDAA   0,X        /HIGH  BYTE Y VALUE
 0468 B2 005A  D               SBCA   Y          /BASE LINE UPPER BYTE
 046B B7 0054  D               STAA   Y1         /THIS IS F(X0)
 046E BD 01BF  P               JSR    NEXPNT     /RIGHT SIDE
 0471 FE 0039  D               LDX    CPTNO
 0474 09                       DEX               Y ADDR
 0475 09                       DEX
 0476 A6 01    A               LDAA   1,X        /LOWER BYTE
 0478 B0 005B  D               SUBA   Y+1        /SUBTRACT  BASE LINE
 047B B7 0057  D               STAA   Y2+1
 047E A6 00    A               LDAA   0,X        /UPPER
 0480 B2 005A  D               SBCA   Y          /BASE LINE UPPER
 0483 B7 0056  D               STAA   Y2         /THIS IS F(X1)
 0486 B6 0055  D               LDAA   Y1+1       /F(X0) LOWER
 0489 BB 0057  D               ADDA   Y2+1       /PLUS F(X1) LOWER
 048C 97 01    B               STAA   MUL1+1
 048E B6 0054  D               LDAA   Y1         /F(X0) UPPER
 0491 B9 0056  D               ADCA   Y2         /PLUS F(X1) UPPER
 0494 97 00    B               STAA   MUL1       /F(X0) + F(X1)
 0496 FE 0039  D               LDX    CPTNO      //CURRENT POINT POINTER
 0499 A6 05    A               LDAA   5,X        //CALC H*2 = X2-X1
 049B A0 01    A               SUBA   1,X
 049D 97 05    B               STAA   MUL3+1
```

```
049F A6 04      A        LDAA   4,X
04A1 A2 00      A        SBCA   0,X
04A3 97 04      B        STAA   MUL3
04A5 74 0004    B        LSR    MUL3      //H = (X2-X1)/2
04A8 76 0005    B        ROR    MUL3+1
04AB BD 0997    P        JSR    MUL       //AREA = H/2*(F(X1)+F(X2))
                  *NOW ADD THIS TRAPEZOID AREA INTO AREA
04AE 96 05      B        LDAA   MUL3+1
04B0 BB 002B    D        ADDA   AREA+3
04B3 B7 002B    D        STAA   AREA+3
04B6 96 04      B        LDAA   MUL3
04B8 B9 002A    D        ADCA   AREA+2
04BB B7 002A    D        STAA   AREA+2
04BE 96 03      B        LDAA   MUL2+1
04C0 B9 0029    D        ADCA   AREA+1
04C3 B7 0029    D        STAA   AREA+1
04C6 96 02      B        LDAA   MUL2
04C8 B9 0028    D        ADCA   AREA
04CB B7 0028    D        STAA   AREA
04CE 39           GTAR3  RTS
                  *
                  * CLEAR THE PLAIN SAMPLE PEAK ARFA BUFFERS
                  *
04CF 86 00      A CLSABF LDAA   #0
04D1 CE 0080    A        LDX    #128
04D4 A7 FB      A CLSA1  STAA   SABF-1,X
04D6 09                  DEX
04D7 26 FB 04D4          BNE    CLSA1
04D9 39                  RTS
                  ********************
                  *SAVE THE INTEGRATION RESULTS IN PROPER BUFFER
                  ********************
04DA B6 0033    D SAVRES LDAA   ELNUM     RSSET ELNUM FOR PROPER
04DD 81 02      A        CMPA   #2        OFFSET
04DF 26 05 04E6          BNE    SAV1
                  *
04E1 7F 0033    D        CLR    ELNUM     SET ZN = 0
04E4 20 16 04FC          BRA    SAV4
                  *
04E6 81 04      A SAV1   CMPA   #4
04E8 26 02 04EC          BNE    SAV2      CD OK
04EA 20 10 04FC          BRA    SAV4
                  *
04EC 81 06      A SAV2   CMPA   #6
04EE 26 07 04F7          BNE    SAV3
                  *
04F0 8B 02      A        ADDA   #2        PB = 8
04F2 B7 0033    D        STAA   ELNUM
04F5 20 05 04FC          BRA    SAV4
                  *
04F7 8B 04      A SAV3   ADDA   #4
04F9 B7 0033    D        STAA   ELNUM     CU = 12
                  *
04FC 7F 0032    D SAV4   CLR    ELNUM1    /HIGH OF INDEX
04FF F6 0000    A        LDAB   CNTR1     //CYCLE #
0502 5A                  DECB             NO 0 CYCLE
0503 58                  ASLB             //*2
0504 58                  ASLB             //*4
```

```
Ø5Ø5 58                ASLB           //*8
Ø5Ø6 58                ASLB           //*16
Ø5Ø7 FB ØØ33  D        ADDB  ELNUM
Ø5ØA F7 ØØ33  D        STAB  ELNUM
Ø5ØD FE ØØ32  D        LDX   ELNUM1
Ø51Ø B6 ØØ28  D        LDAA  AREA
Ø513 A7 FC    A        STAA  SABF,X
Ø515 B6 ØØ29  D        LDAA  AREA+1
Ø518 A7 FD    A        STAA  SABF+1,X
Ø51A B6 ØØ2A  D        LDAA  AREA+2
Ø51D A7 FE    A        STAA  SABF+2,X AREA IS IN 32 BITS
Ø51F B6 ØØ2B  D        LDAA  AREA+3
Ø522 A7 FF    A        STAA  SABF+3,X
Ø524 39                RTS
                *CALCULATE THE CONCENTRATIONS FOR THE THREE
                *PLAIN SAMPLE PEAK AREAS THAT ARE NON-ZERO
                *IN SABF. PUT RESULTS IN CONBF
Ø525 BD Ø551  P CALCON JSR   CLCNBF   / CLEAR CONCENTRATION BUFF
Ø528 7D ØØ97  D        TST   INTCAL
Ø52B 27 Ø8 Ø535        BEQ   CALCØ
                *
Ø52D B6 ØØØØ  A        LDAA  VPRECY
Ø53Ø 4A                DECA
Ø531 4A                DECA
Ø532 B7 ØØ35  D        STAA  SUBSMP   OFFSET
                *
Ø535 86 ØØ    A CALCØ  LDAA  #Ø
Ø537 B7 ØØ33  D CALCO  STAA  ELNUM
Ø53A F6 ØØ35  D        LDAB  SUBSMP
Ø53D 58                ASLB  *2
Ø53E 58                ASLB  *4
Ø53F 58                ASLB  *8
Ø54Ø 58                ASLB  *16
Ø541 F7 ØØ5D  D CALC1  STAB  SABOFF
Ø544 BD Ø565  P        JSR   CL1CON   /TRY TO CALCULATE THIS CON
Ø547 B6 ØØ33  D        LDAA  ELNUM
Ø54A 8B Ø4    A        ADDA  #4
Ø54C 81 ØC    A        CMPA  #12
Ø54E 2F E7 Ø537        BLE   CALCO    /NEXT
Ø55Ø 39                RTS
                *CLEAR THE CONCENTRATION BUFFER
Ø551 4F         CLCNBF CLRA
Ø552 CE ØØ1Ø  A        LDX   #16
Ø555 A7 Ø7    B CLCNB1 STAA  CONBF-1,X
Ø557 Ø9                DEX
Ø558 26 FB Ø555        BNE   CLCNB1
Ø55A 39                RTS
                *
                *CLEAR THE DIVISOR BUFFER (SAAXA)
                *
Ø55B 4F         CLSXA  CLRA
Ø55C CE ØØØ8  A        LDX   #8
Ø55F A7 29    B CLSXA1 STAA  SAAXA-1,X
Ø561 Ø9                DEX
Ø562 26 FB Ø55F        BNE   CLSXA1
Ø564 39                RTS
                *
                *
```

```
                     *CALCULATE CONCENTRATION FOR ONE ELEMENT IF
                     *AREA IS NOT ZERO
                     *
                     *
Ø565 B6 ØØ33    D CL1CON LDAA   ELNUM     CHK IF STD CON IS Ø
                  *                IF IT IS- RETURN
Ø568 16                  TAB              A TO B
Ø569 54                  LSRB             B/2
Ø56A F7 ØØ59    D        STAB   EX+1
Ø56D 7F ØØ58    D        CLR    EX
Ø57Ø FE ØØ58    D        LDX    EX        OFFSET
Ø573 E6 ØØ      A        LDAB   STCN1H,X GET STD CON VALUE
Ø575 5D                  TSTB
Ø576 26 Ø1 Ø579          BNE    CL1C2
Ø578 39                  RTS
                  *
Ø579 B6 ØØ5D    D CL1C2  LDAA   SABOFF    /BUFFER OFFSET
Ø57C BB ØØ33    D        ADDA   ELNUM     /ELEMENT AREA OFFSET
Ø57F B7 ØØ59    D        STAA   EX+1
Ø582 7F ØØ58    D        CLR    EX
Ø585 FE ØØ58    D        LDX    EX
Ø588 6D FC      A        TST    SABF,X
Ø58A 26 ØD Ø599          BNE    CL1C1
Ø58C 6D FD      A        TST    SABF+1,X
Ø58E 26 Ø9 Ø599          BNE    CL1C1
Ø59Ø 6D FE      A        TST    SABF+2,X
Ø592 26 Ø5 Ø599          BNE    CL1C1
Ø594 6D FF      A        TST    SABF+3,X
Ø596 26 Ø1 Ø599          BNE    CL1C1
Ø598 39                  RTS              /NO AREA
Ø599 A6 FC      A CL1C1  LDAA   SABF,X
Ø59B B7 ØØ64    D        STAA   XA
Ø59E A6 FD      A        LDAA   SABF+1,X
Ø5AØ B7 ØØ65    D        STAA   XA+1
Ø5A3 A6 FE      A        LDAA   SABF+2,X
Ø5A5 B7 ØØ66    D        STAA   XA+2
Ø5A8 A6 FF      A        LDAA   SABF+3,X
Ø5AA B7 ØØ67    D        STAA   XA+3      //UNKNOWN PEAK AREA IN XA
                  *
Ø5AD 7D ØØ97    D        TST    INTCAL
Ø5BØ 27 3B Ø5ED          BEQ    XADIV
                  *
Ø5B2 B6 ØØØØ    A        LDAA   VPRECY    //STD ADD CYCLE #
Ø5B5 4A                  DECA             ADJUST TO PROPER POSITION
Ø5B6 48                  ASLA
Ø5B7 48                  ASLA
Ø5B8 48                  ASLA
Ø5B9 48                  ASLA
Ø5BA BB ØØ33    D        ADDA   ELNUM     //CURRENT ELEMENT
Ø5BD B7 ØØ59    D        STAA   EX+1
Ø5CØ 7F ØØ58    D        CLR    EX        OFFSET TO POINT TO
Ø5C3 FE ØØ58    D        LDX    EX        STD PEAK AREA
                  *
Ø5C6 A6 FF      A        LDAA   SABF+3,X USE LO 16 BITS
Ø5C8 BØ ØØ67    D        SUBA   XA+3
Ø5CB B7 ØØ96    D        STAA   SAVA      SAVE DIVISOR
                  *
Ø5CE E6 FE      A        LDAB   SABF+2,X SAA-XA   LO BYTES
```

```
05D0 F2 0066   D       SBCB   XA+2     DIVISOR
05D3 2D 15 05EA        BLT    E1       IF SAA-XA IS NEG: ERROR
                 *
05D5 B6 0033   D       LDAA   ELNUM    GET OFFSET
05D8 44                LSRA            /2
05D9 B7 0059   D       STAA   EX+1
05DC 7F 0058   D       CLR    EX
05DF FE 0058   D       LDX    EX       OFFSET INTO X
05E2 B6 0096   D       LDAA   SAVA     RESTORE A
                 *
05E5 A7 2B     B       STAA   SAAXA+1,X
05E7 E7 2A     B       STAB   SAAXA,X  SAVE DIVISOR'S
05E9 39                RTS
                 *
05EA 7E 06EE   P E1    JMP    ER1      ERROR MESSAGE
                 *
05ED B6 0033   D XADIV LDAA   ELNUM
05F0 44                LSRA
05F1 B7 0059   D       STAA   EX+1
05F4 7F 0058   D       CLR    EX
05F7 FE 0058   D       LDX    EX
                 *
05FA A6 2B     B       LDAA   SAAXA+1,X
05FC E6 2A     B       LDAB   SAAXA,X  DIVISOR FOR CURRENT PEAK
                 *
05FE 5D                TSTB            CHK IF DIVISOR 0
05FF 26 04 0605        BNE    OK
0601 4D                TSTA
0602 26 01 0605        BNE    OK
0604 39                RTS
                 *
0605 FE 0064   D OK    LDX    XA
0608 DF 02     B       STX    DIV2
060A FE 0066   D       LDX    XA+2
060D DF 04     B       STX    DIV3     //XA IS DIVIDEND
                 *
060F BD 09F1   P       JSR    DDIV     // XA/(SAA-XA)
                 *
                 *
                 *   QUOT. = DIV3 & REM = DIV2
                 *    OPERATE ON RESULT SO AS NOT TO LOSE
                 *    REMAINDER FOR NEXT MULTIPLICATION
                 *
                 *
0612 DE 02     B       LDX    DIV2     SAVE REM
0614 FF 006E   D       STX    OR1
                 *
0617 7D 0004   B       TST    DIV3     CHK IF QUOT < 99
061A 26 08 0624        BNE    ER0      IF NOT : ERROR
061C 96 05     B       LDAA   DIV3+1
061E 27 07 0627        BEQ    CLRR1
0620 81 63     A       CMPA   #99
0622 2B 0C 0630        BMI    NOER
0624 7E 06EE   P ER0   JMP    ER1
                 *
0627 CE 0000   A CLRR1 LDX    #0
062A FF 0070   D       STX    R1
062D 7E 064C   P       JMP    OR1A
```

```
                *
0630 CE 0000  A NOER   LDX    #0
0633 DF 02    B        STX    DIV2     DIVIDEND IN DIV2 (HI) & DI
0635 86 0A    A        LDAA   #$0A
0637 5F                CLRB            DIVISOR IN A & B = 10
0638 BD 09F1  P        JSR    DDIV     QUOT/10
                *
063B 96 05    B        LDAA   DIV3+1   LO BYTE QUOT.
063D 48                ASLA
063E 48                ASLA
063F 48                ASLA
0640 48                ASLA
0641 B7 0070  D        STAA   R1       SAVE 1ST DIG IN UPPER
                *                     4 BITS OF R1
                *
0644 96 03    B        LDAA   DIV2+1   LO BYTE REM.
0646 BB 0070  D        ADDA   R1
0649 B7 0070  D        STAA   R1       REM IN LOWER 4 BITS
                *
                *        NOW FORM 2 BCD DIGITS AFTER DEC PT
                *
064C FE 006E  D OR1A   LDX    OR1      GET REM.
064F DF 04    B        STX    MUL3
0651 CE 000A  A        LDX    #10      MULTIPLY BY 10
0654 DF 00    B        STX    MUL1
0656 BD 0997  P        JSR    MUL
                *               PRODUCT: DIV2 (HI) & DIV3 (LO)
0659 FE 0058  D        LDX    EX       GET OFFSET
065C A6 2B    B        LDAA   SAAXA+1,X LO BYTE DIVISOR
065E E6 2A    B        LDAB   SAAXA,X  HI   "     "
0660 BD 09F1  P        JSR    DDIV
                *                       RETURNS: Q-DIV3 & R-DIV2
0663 96 05    B        LDAA   DIV3+1   LO BYTE
0665 48                ASLA
0666 48                ASLA
0667 48                ASLA
0668 48                ASLA            PACK DIGIT INTO UPPER
0669 B7 0072  D        STAA   R2       4 BITS OF R2
                *
066C DE 02    B        LDX    DIV2     GET REMAINDER
066E DF 04    B        STX    MUL3
0670 CE 000A  A        LDX    #10      *10
0673 DF 00    B        STX    MUL1
0675 BD 0997  P        JSR    MUL
                *
0678 FE 0058  D        LDX    EX
067B A6 2B    B        LDAA   SAAXA+1,X (REM*10)/(SAA-XA)
067D E6 2A    B        LDAB   SAAXA,X
067F BD 09F1  P        JSR    DDIV     2ND DIGIT AFTER DP
                *
0682 96 05    B        LDAA   DIV3+1
0684 BB 0072  D        ADDA   R2       FIRST 2 DIGITS AFTER
0687 B7 0072  D        STAA   R2       DEC.PT. IN R2
                *
                *
                *
                *    CONVERT BCD # BACK TO BIN #
                *
```

```
020   COMP    *** DATA COMPUTING ROUTINE ***

 068A B6 0070  D        LDAA    R1        MSD
 068D F6 0072  D        LDAB    R2        LSD
                *
 0690 BD 0000  A        JSR     BCDBIN
                *
 0693 CE 0000  A        LDX     #0
 0696 DF 02    B        STX     MUL2
 0698 B7 0070  D        STAA    R1        HI BYTE MULTIPLICAN
 069B F7 0071  D        STAB    R1+1      LO BYTE
                *
 069E FE 0058  D        LDX     EX
 06A1 EE 00    A        LDX     STCN1H,X GET STD CON. VALUE
 06A3 DF 00    B        STX     MUL1      MULTIPLIER (SAC)
                *
 06A5 FE 0070  D        LDX     R1        GET MULTIPLICAN (XA/(SAA-X
 06A8 DF 04    B        STX     MUL3
 06AA BD 0997  P        JSR     MUL       // (XA/(SAA-XA))*SAC
                *
                *    NOW MUST DIVIDE ANSWER BY 100000 TO
                *      SCALE INTEGER VALUE FOR CORRECT
                *      DEC. PT. POSITION
                *    DIVIDEND IN DIV2 (HI) & DIV3 (LO)
                *
 06AD 86 E8    A        LDAA    #$E8
 06AF C6 03    A        LDAB    #$03      DIVISOR = 1000
 06B1 BD 09F1  P        JSR     DDIV      /1000
                *
 06B4 CE 0000  A        LDX     #0
 06B7 DF 02    B        STX     DIV2      IGNORE REM
 06B9 86 64    A        LDAA    #$64
 06BB 5F                CLRB              DIVISOR = 100
 06BC BD 09F1  P        JSR     DDIV      /100
                *
                *      DIV3 = QUOT.   &   DIV2 = REM
                *
                *
 06BF B6 0033  D        LDAA    ELNUM
 06C2 B7 0059  D        STAA    EX+1
 06C5 7F 0058  D        CLR     EX
 06C8 FE 0058  D        LDX     EX        BUFF. OFFSET
 06CB 96 04    B        LDAA    DIV3
 06CD A7 08    B        STAA    CONBF,X
 06CF 96 05    B        LDAA    DIV3+1
 06D1 A7 09    B        STAA    CONBF+1,X
                *
 06D3 DE 02    B        LDX     DIV2      GET REM
 06D5 DF 04    B        STX     DIV3      SET UP FOR DIV.
 06D7 CE 0000  A        LDX     #0
 06DA DF 02    B        STX     DIV2      CLEAR UPPER 16 BITS
 06DC 86 0A    A        LDAA    #$0A
 06DE 5F                CLRB              DIVISOR =  10
 06DF BD 09F1  P        JSR     DDIV      /10
                *
 06E2 FE 0058  D        LDX     EX
 06E5 96 05    B        LDAA    DIV3+1   LO BYTE QUOT
 06E7 A7 0A    B        STAA    CONBF+2,X
                *
 06E9 96 03    B        LDAA    DIV2+1    STORE REM. TO CONBF+3
```

```
06EB A7 0B    B        STAA   CONBF+3,X
                *
06ED 39                RTS
                *
                *
06EE B6 0033  D ER1    LDAA   ELNUM     CURRENT ELEMENT
06F1 27 0A 06FD        BEQ    Z1
                *
06F3 81 04    A        CMPA   #4
06F5 27 0A 0701        BEQ    CAD
                *
06F7 81 08    A        CMPA   #8
06F9 27 0A 0705        BEQ    LEAD
                *
06FB 20 0C 0709        BRA    COP
                *
06FD 7C 0076  D Z1     INC    Z1F
0700 39                RTS
                *
0701 7C 0077  D CAD    INC    CADF
0704 39                RTS
                *
0705 7C 0079  D LEAD   INC    LEADF
0708 39                RTS
                *
0709 7C 0078  D COP    INC    COPF
070C 39                RTS
                *
                *PRINT CONCENTRATIONS FOR ONE SAMPLE ON THE
                *PRINTER
                *
070D 7F 004D  D PRTSMP CLR    ONES
0710 7F 004E  D        CLR    TENS
0713 7F 004F  D        CLR    HUND
0716 86 01    A        LDAA   #1
0718 B7 0000  A        STAA   BLOCK
                *
071B B6 0036  D        LDAA   SAMPLE    SAMPLE =0?
071E 4D                TSTA             YES: PRINT OUT HEADER MESS
071F 26 0C 072D        BNE    SK30
0721 CE 0000  A        LDX    #MES45
0724 BD 0000  A        JSR    PRINT1
0727 BD 0000  A        JSR    SUB3      LF
072A BD 0000  A        JSR    SUB3
                *
072D 7D 0000  A SK30   TST    ZINK
0730 27 10 0742        BEQ    SK31
                *
0732 7D 0076  D        TST    Z1F       ZN ERROR?
0735 27 19 0750        BEQ    SK32      NO
0737 CE 5A4E  A        LDX    #$5A4E
073A FF 0019  A        STX    MESERR+25
073D 7C 0075  D        INC    ERFLAG
0740 20 0E 0750        BRA    SK32
                *
0742 7D 0078  D SK31   TST    COPF      CU ERROR?
0745 27 09 0750        BEQ    SK32      NO
0747 CE 4355  A        LDX    #$4355    YES
```

```
022   COMP   *** DATA COMPUTING ROUTINE ***

 074A FF 0019  A          STX    MESERR+25
 074D 7C 0075  D          INC    ERFLAG
                 *
 0750 7D 0079  D SK32     TST    LEADF     PB ERROR?
 0753 27 09 075E          BEQ    SK33      NO
 0755 CE 5042  A          LDX    #$5042    YES
 0758 FF 001C  A          STX    MESERR+28
 075B 7C 0075  D          INC    ERFLAG
                 *
 075E 7D 0077  D SK33     TST    CADF      CD ERROR?
 0761 27 09 076C          BEQ    SK34      NO
 0763 CE 4344  A          LDX    #$4344    YES
 0766 FF 001F  A          STX    MESERR+31
 0769 7C 0075  D          INC    ERFLAG
                 *
 076C 7D 0075  D SK34     TST    ERFLAG
 076F 27 15 0786          BEQ    SK35
                 *
 0771 CE 0000  A          LDX    #MESERR   IF THERE WAS AN
 0774 BD 0000  A          JSR    PRINT1    ERROR, PRINT MESG
 0777 7F 0076  D          CLR    Z1F
 077A 7F 0078  D          CLR    COPF
 077D 7F 0077  D          CLR    CADF      CLEAR ALL FLAGS
 0780 7F 0079  D          CLR    LEADF
 0783 7F 0075  D          CLR    ERFLAG
                 *
 0786 BD 0796  P SK35     JSR    CLPRBF
 0789 BD 07A4  P          JSR    PRDATE    PRINT DATE LINE
 078C BD 0796  P          JSR    CLPRBF
 078F BD 081F  P          JSR    PRCONS    CONC. LINE
 0792 BD 0000  A          JSR    SUB3
 0795 39                  RTS
                 *
 0796 C6 28    A CLPRBF LDAB     #40       CLEAR PRINT BUFFER
 0798 86 20    A          LDAA   #$20
 079A CE 0000  D          LDX    #PRBUF
 079D A7 00    A CLPR1    STAA   0,X
 079F 08                  INX
 07A0 5A                  DECB
 07A1 26 FA 079D          BNE    CLPR1
 07A3 39                  RTS
                 *
                 *
                 * PRINT THE SAMPLE NUMBER
                 *   AND THE DATE AND TIME
                 *
 07A4 CE 0000  D PRDATE LDX      #PRBUF    /ADDRESS OF PRINT BUFFER
 07A7 86 2A    A          LDAA   #'*
 07A9 A7 00    A          STAA   0,X
 07AB 08                  INX
 07AC F6 0036  D          LDAB   SAMPLE    /SAMPLE #
 07AF 5C                  INCB             NO 0 SAMPLE #
 07B0 4F                  CLRA   INC RESUP /HIGH BYTE IS 0
 07B1 7C 004F  D      +   INC    HUND
 07B4 BD 0910  P          JSR    BINTOA    /CORRECT BINARY TO ASCII
 07B7 7F 004F  D          CLR    HUND
 07BA CE 0008  D          LDX    #PRBUF+8 /START OF DATA
 07BD F6 0000  A          LDAB   YRL       /YEAR LOW BYTE
```

```
07C0 B6 0000  A         LDAA    YRH       /YEAR HIGH BYTE
07C3 7C 0074  D         ~~INC     ZESUP~~     DON'T ZERO SUPRES
07C6 BD 0910  P         JSR     BINTOA    /CONVERT & PUT IN BUF
07C9 08                 INX               LEAVE 2 BLANKS
07CA 08                 INX
07CB 7C 004F  D         INC     HUND
07CE F6 0000  A         LDAB    DAYL      /DAY OF THE YEAR
07D1 B6 0000  A         LDAA    DAYH
07D4 BD 0910  P         JSR     BINTOA    /CONVERT & PUT IN BUF
07D7 7F 004F  D         CLR     HUND
07DA 08                 INX               /2 BLANKS
07DB 08                 INX
07DC FF 0094  D         STX     SAVX      SAVE CURRENT X
07DF FE 0092  D         LDX     TMBUF     GET SAMPLE AQUIS. TIME
07E2 A6 00    A         LDAA    0,X
07E4 08                 INX
07E5 E6 00    A         LDAB    0,X
07E7 08                 INX
07E8 FF 0092  D         STX     TMBUF     UPDATE POINTER
07EB FE 0094  D         LDX     SAVX      RESET X
                  *
07EE BD 0000  A         JSR     BCDBIN    CONV. BCD TO BIN
                  *
07F1 BD 0910  P         JSR     BINTOA    /CONVERT & PUT IN BUF
                  *
07F4 86 3A    A         LDAA    #':
07F6 A7 00    A         STAA    0,X
07F8 08                 INX
07F9 4F                 CLRA
07FA FF 0094  D         STX     SAVX
07FD FE 0092  D         LDX     TMBUF
0800 E6 00    A         LDAB    0,X
0802 08                 INX
0803 FF 0092  D         STX     TMBUF
0806 FE 0094  D         LDX     SAVX
0809 7C 004E  D         INC     TENS
080C BD 0000  A         JSR     BCDBIN
080F BD 0910  P         JSR     BINTOA
0812 7F 004E  D         CLR     TENS
                  *
0815 CE 0000  D         LDX     #PRBUF
0818 BD 0000  A         JSR     PRINT1
081B 7F 0074  D         CLR     ZESUP
081E 39                 RTS
                  *
                  *PRINT THE CONCENTRATION LINE
                  *
081F 7D 0000  A PRCONS  TST     ZINK      LOOKING FOR ZINC?
0822 27 3A 085E         BEQ     PNT1      NO: SKIP ZN
                  *
0824 CE 0005  D         LDX     #PRBUF+5 FILL BUFFER
0827 86 5A    A         LDAA    #'Z
0829 A7 00    A         STAA    0,X
082B 08                 INX
082C 86 4E    A         LDAA    #'N
082E A7 00    A         STAA    0,X
0830 08                 INX
0831 86 3A    A         LDAA    #':
```

```
Ø833 A7 ØØ     A           STAA   Ø,X
Ø835 Ø8                    INX
                *
Ø836 7C ØØ4F   D           INC    HUND      # <= 999
Ø839 96 Ø8     B           LDAA   CONBF     ZN CONC. VAL
Ø83B D6 Ø9     B           LDAB   CONBF+1
                *
Ø83D BD Ø91Ø   P           JSR    BINTOA
Ø84Ø BD Ø9ØA   P           JSR    DECPT
                *
Ø843 D6 ØA     B           LDAB   CONBF+2   # AFTER DEC PT
Ø845 4F                    CLRA
Ø846 7C ØØ4D   D           INC    ONES
Ø849 7C ØØ74   D           INC    ZESUP     DON'T SUPRESS Ø'S
                *
Ø84C BD Ø91Ø   P           JSR    BINTOA
                *
Ø84F D6 ØB     B           LDAB   CONBF+3
Ø851 4F                    CLRA
Ø852 BD Ø91Ø   P           JSR    BINTOA
Ø855 7F ØØ4D   D           CLR    ONES
Ø858 7F ØØ74   D           CLR    ZESUP
Ø85B 7E Ø895   P           JMP    PNT2
                *
Ø85E CE ØØØ5   D PNT1      LDX    #PRBUF+5
Ø861 86 43     A           LDAA   #'C
Ø863 A7 ØØ     A           STAA   Ø,X
Ø865 Ø8                    INX
Ø866 86 55     A           LDAA   #'U
Ø868 A7 ØØ     A           STAA   Ø,X
Ø86A Ø8                    INX
Ø86B 86 3A     A           LDAA   #':
Ø86D A7 ØØ     A           STAA   Ø,X
Ø86F Ø8                    INX
                *
Ø87Ø 7C ØØ4F   D           INC    HUND
Ø873 96 14     B           LDAA   CONBF+12 CU
Ø875 D6 15     B           LDAB   CONBF+13
Ø877 BD Ø91Ø   P           JSR    BINTOA
Ø87A BD Ø9ØA   P           JSR    DECPT
                *
Ø87D D6 16     B           LDAB   CONBF+14
Ø87F 4F                    CLRA
Ø88Ø 7C ØØ4D   D           INC    ONES
Ø883 7C ØØ74   D           INC    ZESUP
Ø886 BD Ø91Ø   P           JSR    BINTOA
                *
Ø889 D6 17     B           LDAB   CONBF+15
Ø88B 4F                    CLRA
Ø88C BD Ø91Ø   P           JSR    BINTOA
Ø88F 7F ØØ4D   D           CLR    ONES
Ø892 7F ØØ74   D           CLR    ZESUP
                *
Ø895 CE ØØ11   D PNT2      LDX    #PRBUF+17
Ø898 86 5Ø     A           LDAA   #'P
Ø89A A7 ØØ     A           STAA   Ø,X
Ø89C Ø8                    INX
Ø89D 86 42     A           LDAA   #'B
```

```
025    COMP    *** DATA COMPUTING ROUTINE ***

089F A7 00      A             STAA   0,X
08A1 08                       INX
08A2 86 3A      A             LDAA   #':
08A4 A7 00      A             STAA   0,X
08A6 08                       INX
                  *
08A7 96 10      B             LDAA   CONBF+8
08A9 D6 11      B             LDAB   CONBF+9
08AB BD 0910    P             JSR    BINTOA
08AE BD 090A    P             JSR    DECPT
                  *
08B1 D6 12      B             LDAB   CONBF+10
08B3 4F                       CLRA
08B4 7C 004D    D             INC    ONES
08B7 7C 0074    D             INC    ZESUP
08BA BD 0910    P             JSR    BINTOA
                  *
08BD D6 13      B             LDAB   CONBF+11
08BF 4F                       CLRA
08C0 BD 0910    P             JSR    BINTOA
08C3 7F 004D    D             CLR    ONES
08C6 7F 0074    D             CLR    ZESUP
                  *
08C9 CE 001D    D             LDX    #PRBUF+29
08CC 86 43      A             LDAA   #'C
08CE A7 00      A             STAA   0,X
08D0 08                       INX
08D1 86 44      A             LDAA   #'D
08D3 A7 00      A             STAA   0,X
08D5 08                       INX
08D6 86 3A      A             LDAA   #':
08D8 A7 00      A             STAA   0,X
08DA 08                       INX
                  *
08DB 96 0C      B             LDAA   CONBF+4    CD
08DD D6 0D      B             LDAB   CONBF+5
08DF BD 0910    P             JSR    BINTOA
08E2 BD 090A    P             JSR    DECPT
08E5 7F 004F    D             CLR    HUND
                  *
08E8 D6 0E      B             LDAB   CONBF+6
08EA 4F                       CLRA
08EB 7C 004D    D             INC    ONES
08EE 7C 0074    D             INC    ZESUP
08F1 BD 0910    P             JSR    BINTOA
                  *
08F4 D6 0F      B             LDAB   CONBF+7
08F6 4F                       CLRA
08F7 BD 0910    P             JSR    BINTOA
08FA 7F 004D    D             CLR    ONES
08FD 7F 0074    D             CLR    ZESUP
                  *
                  *
0900 CE 0000    D             LDX    #PRBUF
0903 BD 0000    A             JSR    PRINT1
0906 BD 0000    A             JSR    SUB3
0909 39                       RTS
                  *
```

```
                         *
                         *
                         * PUT A DECIMAL POINT IN THE CONCENTRATION
                         *
090A 86 2E       A DECPT   LDAA   #'.
090C A7 00       A         STAA   0,X
090E 08                    INX
090F 39                    RTS
                         *
                         *
                         *CONVERT THE NUMBER IN A AND B TO ASCII
                         *AND PUT IT IN THE MEMORY THAT IS POINTED TO BY X
          0004   B DVQUO   EQU    MUL3
          0002   B DVREM   EQU    MUL2
0910 FF 002C     D BINTOA  STX    BINTP1
0913 B7 002E     D         STAA   BINTP2
0916 F7 002F     D         STAB   BINTP2+1
0919 FE 002E     D         LDX    BINTP2
091C 86 0A       A         LDAA   #10
091E 5F                    CLRB
091F BD 09EF     P         JSR    DIV       /DIVIDE BY 10
0922 96 03       B         LDAA   DVREM+1   /REMAINDER
0924 B7 0061     D         STAA   BIONES    /ONES DIGIT
                         *
0927 B6 004D     D         LDAA   ONES
092A 4D                    TSTA
092B 27 06 0933            BEQ    SK20
092D FE 002C     D         LDX    BINTP1
0930 7E 0981     P         JMP    ONE1
                         *
0933 DE 04       B SK20    LDX    DVQUO
0935 86 0A       A         LDAA   #10
0937 5F                    CLRB
0938 BD 09EF     P         JSR    DIV
093B 96 03       B         LDAA   DVREM+1   /REMAINDER
093D B7 0060     D         STAA   BITENS    /TENS DIGIT
                         *
0940 B6 004E     D         LDAA   TENS
0943 4D                    TSTA
0944 27 06 094C            BEQ    SK21
0946 FE 002C     D         LDX    BINTP1
0949 7E 097C     P         JMP    TEN1
                         *
094C DE 04       B SK21    LDX    DVQUO
094E 86 0A       A         LDAA   #10
0950 5F                    CLRB
0951 BD 09EF     P         JSR    DIV
0954 96 03       B         LDAA   DVREM+1
0956 B7 005F     D         STAA   BIHUND    /HUNDREDS DIGIT
                         *
0959 B6 004F     D         LDAA   HUND
095C 4D                    TSTA
095D 27 06 0965            BEQ    SK22
095F FE 002C     D         LDX    BINTP1
0962 7E 0977     P         JMP    HUN1
                         *
0965 DE 04       B SK22    LDX    DVQUO
0967 86 0A       A         LDAA   #10
```

```
Ø969 5F                   CLRB
Ø96A BD Ø9EF  P           JSR    DIV
Ø96D 96 Ø3    B           LDAA   DVREM+1
Ø96F B7 ØØ5E  D           STAA   BITHOU     /THOUSANDS DIGIT
Ø972 FE ØØ2C  D           LDX    BINTP1
Ø975 8D 1Ø Ø987           BSR    CVTDGT
Ø977 B6 ØØ5F  D HUN1      LDAA   BIHUND
Ø97A 8D ØB Ø987           BSR    CVTDGT
Ø97C B6 ØØ6Ø  D TEN1      LDAA   BITENS
Ø97F 8D Ø6 Ø987           BSR    CVTDGT
Ø981 B6 ØØ61  D ONE1      LDAA   BIONES
Ø984 8D Ø1 Ø987           BSR    CVTDGT     /ALWAYS CONVERT LAST DIGIT
Ø986 39                   RTS
                *
Ø987 7D ØØ74  D CVTDGT TST    ZESUP      IF SET: DON'T
Ø98A 26 Ø3 Ø98F           BNE    CVTDG2     SUPRESS Ø'S
Ø98C 4D                   TSTA              ZERO SUPRESION
Ø98D 27 Ø6 Ø995           BEQ    CVTDG1
Ø98F 8B 3Ø    A CVTDG2 ADDA   #$3Ø       / MAKE ASCII #
Ø991 A7 ØØ    A           STAA   Ø,X
Ø993 Ø8                   INX
Ø994 39                   RTS
Ø995 Ø8         CVTDG1 INX
Ø996 39                   RTS
                *
                *
                *MULTIPLY TWO 16 BIT SIGNED NUMBERS YIELDING A 32
                *NUMBER. CALL WITH*
                *MULTIPLIER IN MUL1
                *MULTIPLICAND IN MUL3
                *RETURN WITH:
                *PRODUCT HIGH 16 BITS IN MUL2
                *PRODUCT LOW 16 BITS IN MUL3
                *
                *
                *
Ø997 7F ØØ62  D MUL       CLR    RETF
Ø99A DE Ø4    B           LDX    MUL3
Ø99C DF Ø6    B           STX    MUL4
Ø99E CE ØØØ4  A           LDX    #4
Ø9A1 4F                   CLRA
Ø9A2 A7 Ø1    B LP1       STAA   MUL1+1,X
Ø9A4 Ø9                   DEX
Ø9A5 26 FB Ø9A2           BNE    LP1        //CLEAR WORKING REGISTERS
Ø9A7 CE ØØ1Ø  A           LDX    #16        //SET SHIFT COUNT TO 16
Ø9AA 96 Ø1    B LP2       LDAA   MUL1+1
Ø9AC 84 Ø1    A           ANDA   #1         //GET Y(LSBIT)
Ø9AE 16                   TAB               //SAVE Y(LSBIT) IN ACCB
Ø9AF B8 ØØ62  D           EORA   RETF       //Y(LSBIT)=Y(LSBIT-1) ?
Ø9B2 27 1D Ø9D1           BEQ    SHIFT      //YES: GO TO SHIFT ROUTINE
Ø9B4 5D                   TSTB              //NO: DOES Y(LSBIT) = Ø ?
Ø9B5 27 ØE Ø9C5           BEQ    ADD        //YES: GO TO ADD ROUTINE
Ø9B7 96 Ø3    B           LDAA   MUL2+1     //NO: SUBTRACT MULTIPLICAN
Ø9B9 D6 Ø2    B           LDAB   MUL2       //FROM THE PRODUCT WITH TH
Ø9BB 9Ø Ø7    B           SUBA   MUL4+1     //MSBYTES LINED UP,
Ø9BD D2 Ø6    B           SBCB   MUL4
Ø9BF 97 Ø3    B           STAA   MUL2+1
Ø9C1 D7 Ø2    B           STAB   MUL2
```

```
 09C3 20 0C 09D1         BRA    SHIFT     //THEN GO TO SHIFT ROUTINE
 09C5 96 03     B ADD    LDAA   MUL2+1    //ADD MULITPLICAND TO THE
 09C7 D6 02     B        LDAB   MUL2      //PRODUCT WITH THE MSBYTES
 09C9 9B 07     B        ADDA   MUL4+1    //LINED UP.
 09CB D9 06     B        ADCB   MUL4
 09CD 97 03     B        STAA   MUL2+1
 09CF D7 02     B        STAB   MUL2
 09D1 7F 0062   D SHIFT  CLR    RETF      //CLEAR TEST BYTE
 09D4 76 0000   B        ROR    MUL1      //SHIFT THE MULTIPLIER RIG
 09D7 76 0001   B        ROR    MUL1+1    //ONE BIT SHIFTING THE LSP
 09DA 79 0062   D        ROL    RETF      //INTO THE LSBIT OF FF.
 09DD 77 0002   B        ASR    MUL2      //SHIFT THE PRODUCT RIGHT
 09E0 76 0003   B        ROR    MUL2+1    //BIT, THE MSB REMAINING T
 09E3 76 0004   B        ROR    MUL3      //SAME.
 09E6 76 0005   B        ROR    MUL3+1
 09E9 09                 DEX              //DECREMENT THE SHIFT COUN
 09EA 26 BE 09AA         BNE    LP2       //IF NOT 0 CONTINUE
 09EC DE 04     B        LDX    MUL3      //RETURN PRODUCT LOWER 16
 09EE 39                 RTS
                  *
                  *DIV-CALL WITH:
                  *16 BIT DIVIDEND IN X
                  *16 BIT DIVISOR IN A(LOW) AND B(HIGH)
                  *
                  * DDIV - DOUBLE DIVIDE - CALL WITH:
                  *   32 BIT DIVIDEND IN DIV2 AND DIV3
                  *   DIVISOR LOW IN A REG
                  *   DIVISOR HI IN B REG
                  *
                  *
                  *RETURNS:
                  *  16 BIT QUOTIENT IN DIV3
                  *  16 BIT REMAINDER IN DIV2
                  ******************************
 09EF 8D 63 0A54  DIV    BSR    STOPDS    //DIVIDEND LOW
 09F1 7F 0062   D DDIV   CLR    RETF      //RETURN QUOTIENT
 09F4 97 01     B LD9    STAA   DIV1+1    //DIVISOR LOW
 09F6 D7 00     B        STAB   DIV1      //DIVISOR HIGH
 09F8 7F 0063   D        CLR    SIGN      //FLAG FOR SIGN OF RESULTS
 09FB 7D 0000   B        TST    DIV1      //DIVISOR SIGN
 09FE 2A 07 0A07         BPL    LD1       //ITS POSITIVE
 0A00 CE 0001   B        LDX    #DIV1+1   //LOW BYTE OF DIVISOR
 0A03 C6 02     A        LDAB   #2        //TWO BYTES LONG
 0A05 8D 69 0A70         BSR    COMP      //MAKE IT POSITIVE
 0A07 7D 0002   B LD1    TST    DIV2      //SIGN OF DIVIDEND
 0A0A 2A 04 0A10         BPL    L5
 0A0C C6 04     A        LDAB   #4        //4 BYTES LONG
 0A0E 8D 5D 0A6D         BSR    COMPF     //MAKE IT POSITIVE
 0A10 BD 0A8F   P L5     JSR    OVFTST    //TEST FOR OVERFLOW
 0A13 CE 0011   A        LDX    #17       //# TIMES THRU MAIN LOOP
 0A16 5F          L0     CLRB             //QUOTIENT BIT IN B
 0A17 96 00     B        LDAA   DIV1      //DIVISOR UPPER
 0A19 91 02     B        CMPA   DIV2      //DIVIDEND HIGH UPPFR
 0A1B 27 62 0A7F         BEQ    L2        //EQUAL-TEST LOWER BYTE
 0A1D 22 0D 0A2C         BHI    L1        //DIVISOR > DIVIDEND
 0A1F 5C          L3     INCB             //DIVISOR <= DIVIDEND
 0A20 96 03     B        LDAA   DIV2+1    //DIVIDEND HIGH LOWER
 0A22 90 01     B        SUBA   DIV1+1    //DIVISOR LOWER
```

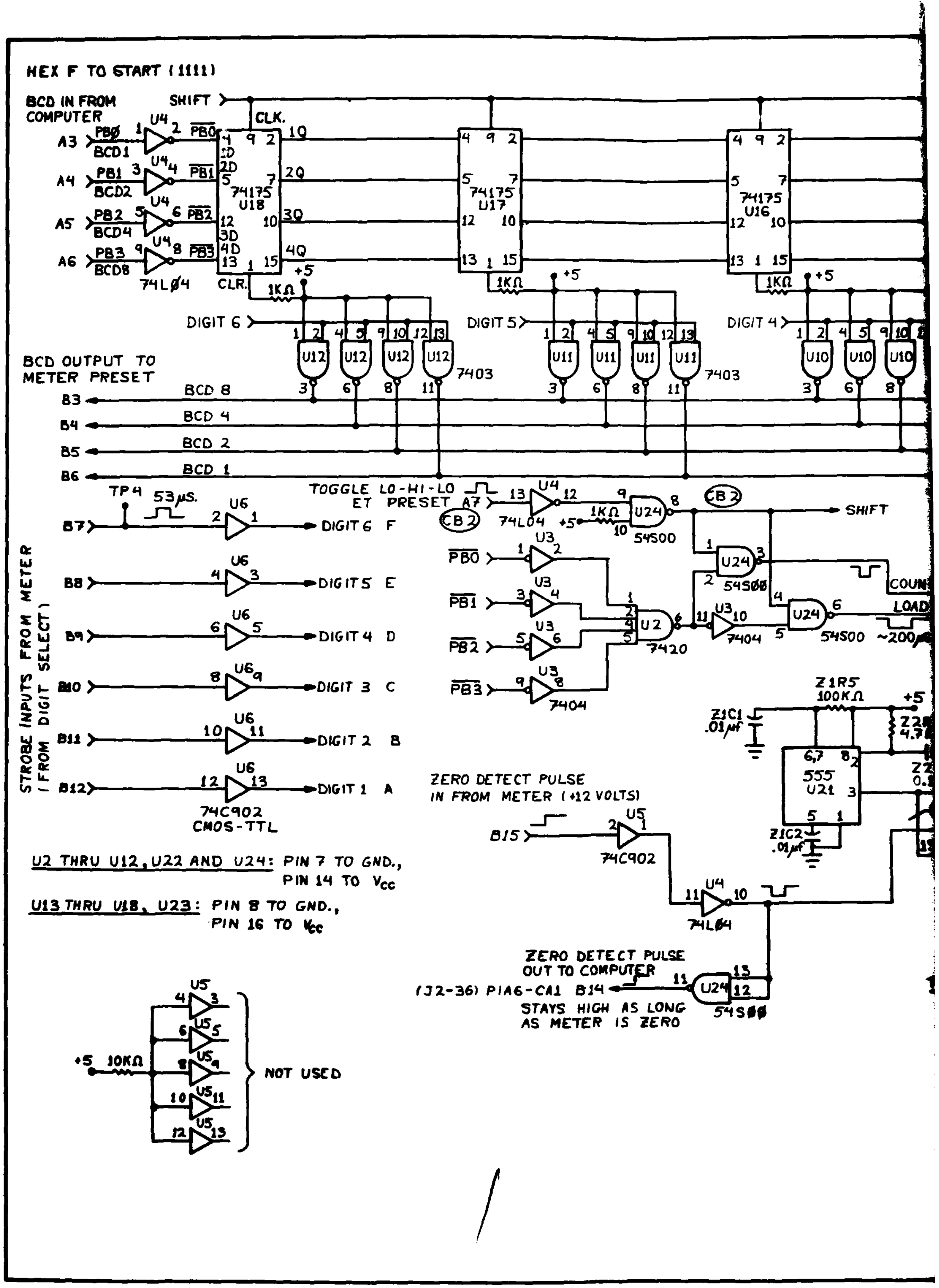
```
ØA24 97 Ø3     B        STAA  DIV2+1
ØA26 96 Ø2     B        LDAA  DIV2      //DIVIDEND HIGH UPPFR
ØA28 92 ØØ     B        SBCA  DIV1      //DIVISOR UPPER
ØA2A 97 Ø2     B        STAA  DIV2
ØA2C 8C ØØØ1   A L1     CPX   #1        //LAST TIME THRU
ØA2F 27 56 ØA87         BEQ   L7
ØA31 8D 6F ØAA2         BSR   STSHF     //SHIFT DIVIDEND LEFT ONE
ØA33 79 ØØØ3   B        ROL   DIV2+1
ØA36 79 ØØØ2   B        ROL   DIV2
ØA39 DA Ø5     B L6     ORAB  DIV3+1    //PUT IN QUOTIENT BIT
ØA3B D7 Ø5     B        STAB  DIV3+1
ØA3D Ø9                 DEX
ØA3E 26 D6 ØA16         BNE   LØ        //DO LOOP 17 TIMES
ØA4Ø 86 Ø1     A        LDAA  #1
ØA42 B4 ØØ63   D        ANDA  SIGN      //MAY NEED TO COMPLEMENT R
ØA45 27 Ø4 ØA4B         BEQ   L8
ØA47 C6 Ø2     A        LDAB  #2
ØA49 8D 22 ØA6D         BSR   COMPF     //COMPLEMENT QUOTIENT
ØA4B 7D ØØ62   D L8     TST   RETF      //RETURN FLAG
ØA4E 27 3B ØA8B         BEQ   L4        //RETURN QUOTIENT
ØA5Ø DE Ø2     B        LDX   DIV2      //RETURN REMAINDER
ØA52 ØC                 CLC             //NO OVERFLOW
ØA53 39                 RTS
ØA54 DF Ø4     B STOPDS STX   DIV3      //DIVIDEND LOW
ØA56 CE ØØØØ   A        LDX   #Ø
ØA59 7D ØØØ4   B        TST   DIV3      //SIGN BIT
ØA5C 2A Ø1 ØA5F         BPL   STOP1     //ITS POSITIVE
ØA5E Ø9                 DEX             //UPPER WORD IS ALL ONES
ØA5F DF Ø2     B STOP1  STX   DIV2      //DIVIDEND HIGH
ØA61 39                 RTS
ØA62 8D FØ ØA54  REM    BSR   STOPDS    //STORE OPERAND IN X
ØA64 CE FFFF   A        LDX   #$FFFF
ØA67 FF ØØ62   D        STX   RETF      //RETURN REMAINDER
ØA6A 7E Ø9F4   P        JMP   LD9
ØA6D CE ØØØ5   B COMPF  LDX   #DIV3+1   //DIVIDEND/QUOTIENT LOWER
ØA7Ø ØC          COMP   CLC             //COMPLEMENT NUM POINTED T
ØA71 86 ØØ     A CML1   LDAA  #Ø
ØA73 A2 ØØ     A        SBCA  Ø,X       //COMPLEMENT BYTE
ØA75 A7 ØØ     A        STAA  Ø,X       //STORE IT BACK
ØA77 Ø9                 DEX
ØA78 5A                 DECB            //BYTE COUNTER
ØA79 26 F6 ØA71         BNE   CML1
ØA7B 7C ØØ63   D        INC   SIGN      //CHANGED ONE ADDRESS
ØA7E 39                 RTS
ØA7F 96 Ø1     B L2     LDAA  DIV1+1    //DIVISOR LOWER
ØA81 91 Ø3     B        CMPA  DIV2+1    //DIVIDEND HIGH LOWER
ØA83 22 A7 ØA2C         BHI   L1        //QUOTIENT BIT IS Ø
ØA85 2Ø 98 ØA1F         BRA   L3        //QUOTIENT BIT IS 1
ØA87 8D 19 ØAA2  L7     BSR   STSHF     //SHIFT ONLY QUOTIENT LAST
ØA89 2Ø AE ØA39         BRA   L6
ØA8B DE Ø4     B L4     LDX   DIV3      //QOTIENT
ØA8D ØC                 CLC             //NO OVERFLOW
ØA8E 39                 RTS
ØA8F 96 Ø2     B OVFTST LDAA  DIV2      //DIVIDEND HIGH UPPER
ØA91 91 ØØ     B        CMPA  DIV1      //DIVISOR UPPER
ØA93 2E Ø9 ØA9E         BGT.  OVFYES
ØA95 2D Ø6 ØA9D         BLT   OVFNO
ØA97 96 Ø1     B        LDAA  DIV1+1    //DIVISOR LOWER
```

```
030   COMP    *** DATA COMPUTING ROUTINE ***

 0A99 91 03     B         CMPA    DIV2+1     //DIVIDEND HIGH LOWER
 0A9B 23 01 0A9E          BLS     OVFYES
 0A9D 39          OVFNO   RTS
 0A9E 32          OVFYES  PULA               //REMOVE OUR RETURN ADDR
 0A9F 32                  PULA
 0AA0 0D                  SEC                //SET OVERFLOW FLAG
 0AA1 39                  RTS                //RETURN TO DIV CALLER
 0AA2 78 0005   B STSHF   ASL     DIV3+1
 0AA5 79 0004   B         ROL     DIV3
 0AA8 39                  RTS
                  * 16 BIT SIGNED COMPARE ROUTINE
                  *  COMPARE X TO A (HGIH) AND B (LOW)
                  *  RETURN (IN B):
                  *    -1 IF X < AB
                  *     0 IF X = AB
                  *     1 IF X > AB
                  *
 0AA9 FF 0030   D CMP16   STX     TPC16      //TEMP
 0AAC B1 0030   D         CMPA    TPC16      //COMPARE HIGH BYTE
 0AAF 2D 0B 0ABC          BLT     CMP1       // X > AB
 0AB1 2E 0C 0ABF          BGT     CMM1       // X < AB
 0AB3 F1 0031   D         CMPB    TPC16+1    //COMPARE LOW BYTE
 0AB6 22 07 0ABF          BHI     CMM1       // X < AB
 0AB8 25 02 0ABC          BCS     CMP1       // X > AB
 0ABA 5F                  CLRB               // X = AB
 0ABB 39                  RTS
 0ABC C6 01     A CMP1    LDAB    #1         // X > AB
 0ABE 39                  RTS
 0ABF C6 FF     A CMM1    LDAB    #$FF       // X < AB
 0AC1 39                  RTS
                  *
                  *
                  *
                          XREF    VPRECY,CNTR2,DAYH,DAYL,HR,MIN
                          XREF    SEC,CNTR1,DATA,YRH,YRL,BCDBIN
                          XREF    STCN1H,BLOCK,PRINT1,SUB3,ZINK
                          XREF    MES45,MESERR,TDATA
                  *
                          XDEF    COMPT,LOWECO,HIECO,AREA,SAMPLE
                          XDEF    CONBF,MUL1,ERFLAG,TIMBUF,TMBUF
                          XDEF    PRBUF,FPTNO,LPTNO,CLSABF
                  *
                          END
 ERRORS 00000
```

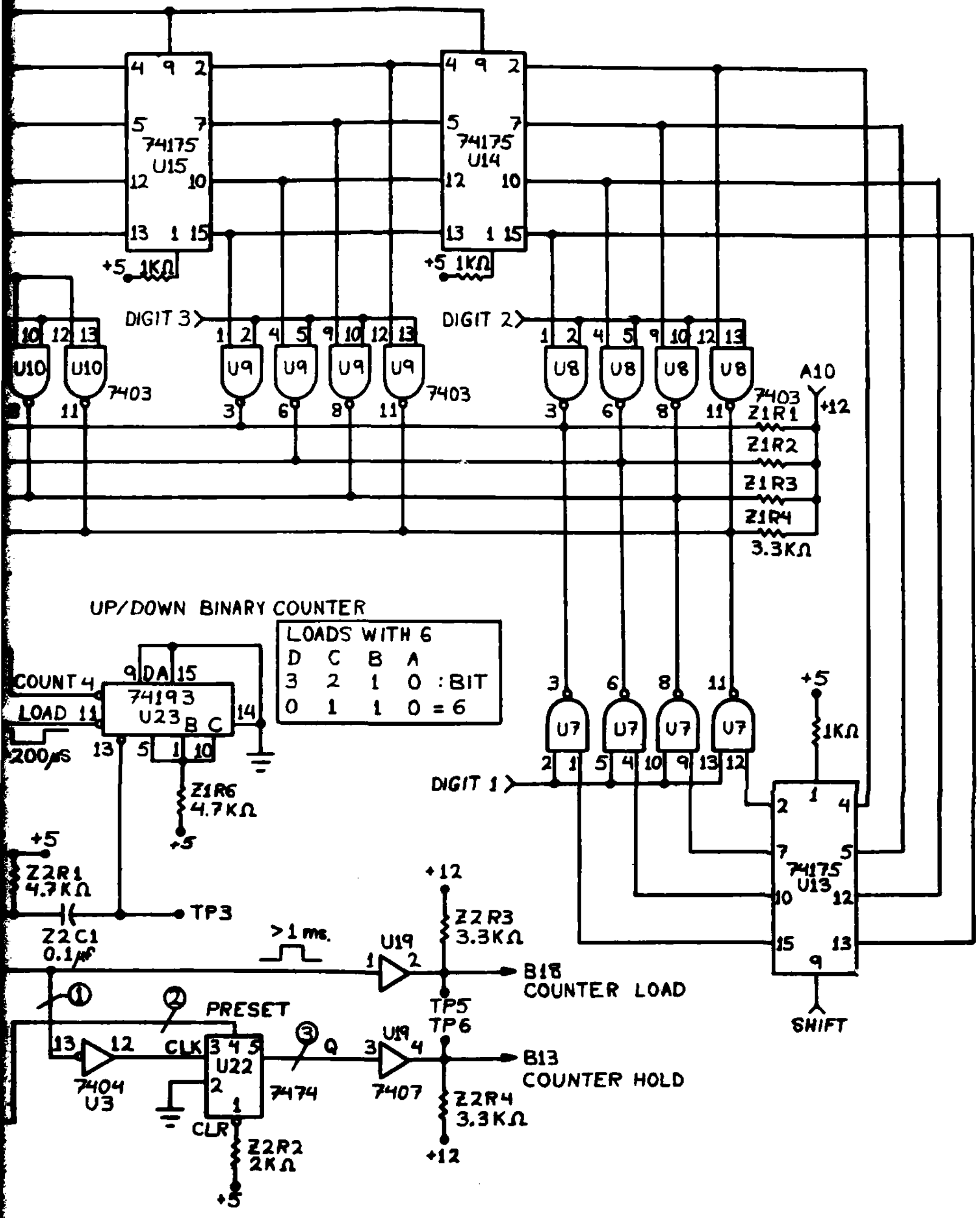
# INDEX TO APPENDIX B

## SYSTEM SCHEMATICS

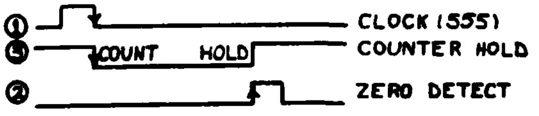


U2 THRU U12, U22 AND U24: PIN 7 TO GND., PIN 14 TO $V_{cc}$

U13 THRU U18, U23: PIN 8 TO GND., PIN 16 TO $V_{cc}$



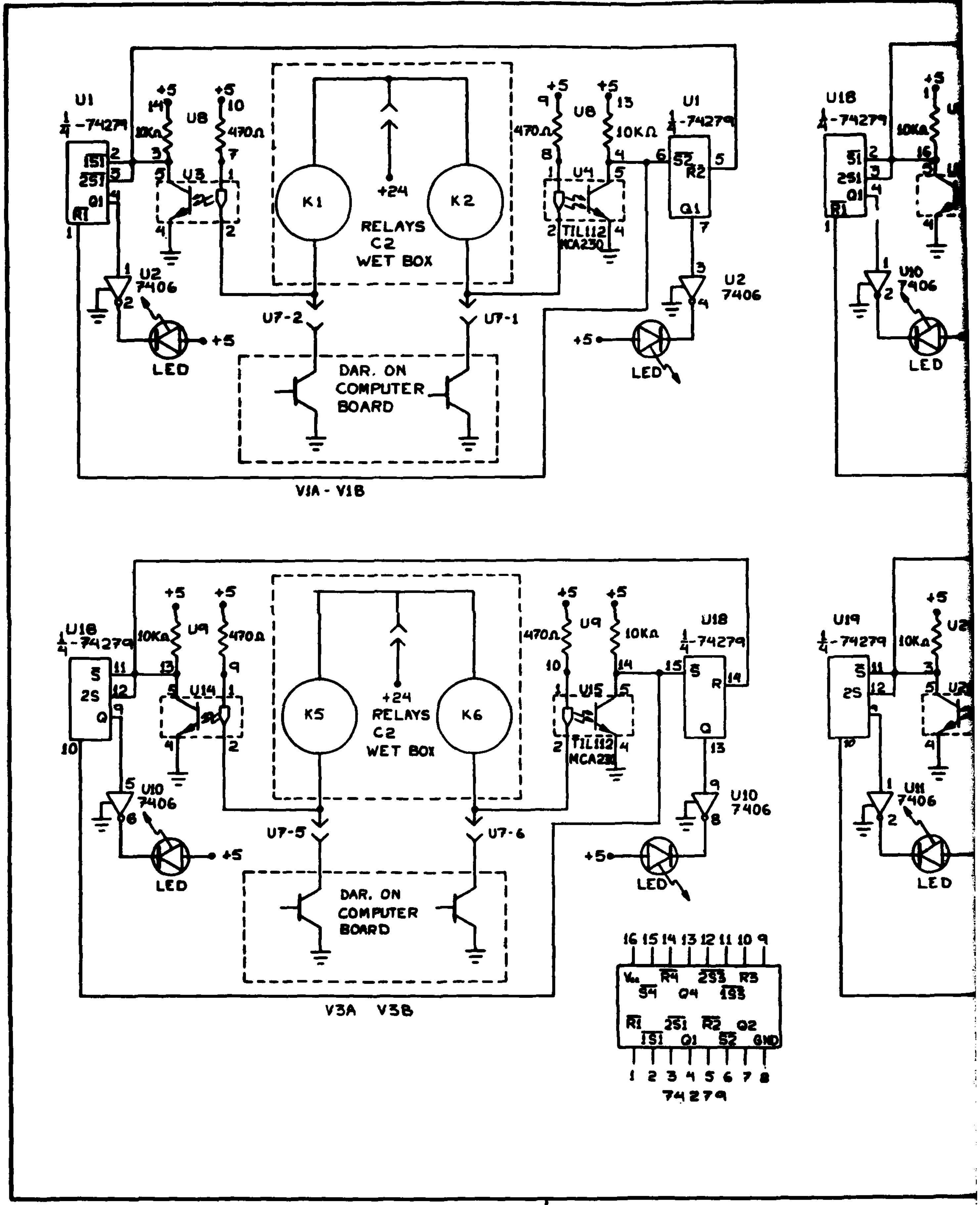
| LOADS WITH 6 | | | | |
|---|---|---|---|---|
| D | C | B | A | |
| 3 | 2 | 1 | 0 | : BIT |
| 0 | 1 | 1 | 0 | = 6 |



TO USE, ENTER:

1) ØF∞
2) TOGGLE SHIFT (CB 2)
3) DUMMY DIGIT
4) TOGGLE SHIFT (TS)
5) 1ST DIGIT
TS
6) 2ND DIGIT
TS
7) 3RD DIGIT
TS
8) 4TH DIGIT
TS
9) 5TH DIGIT
TS
10) 6TH DIGIT
TS; WILL LOAD & START COUNTING

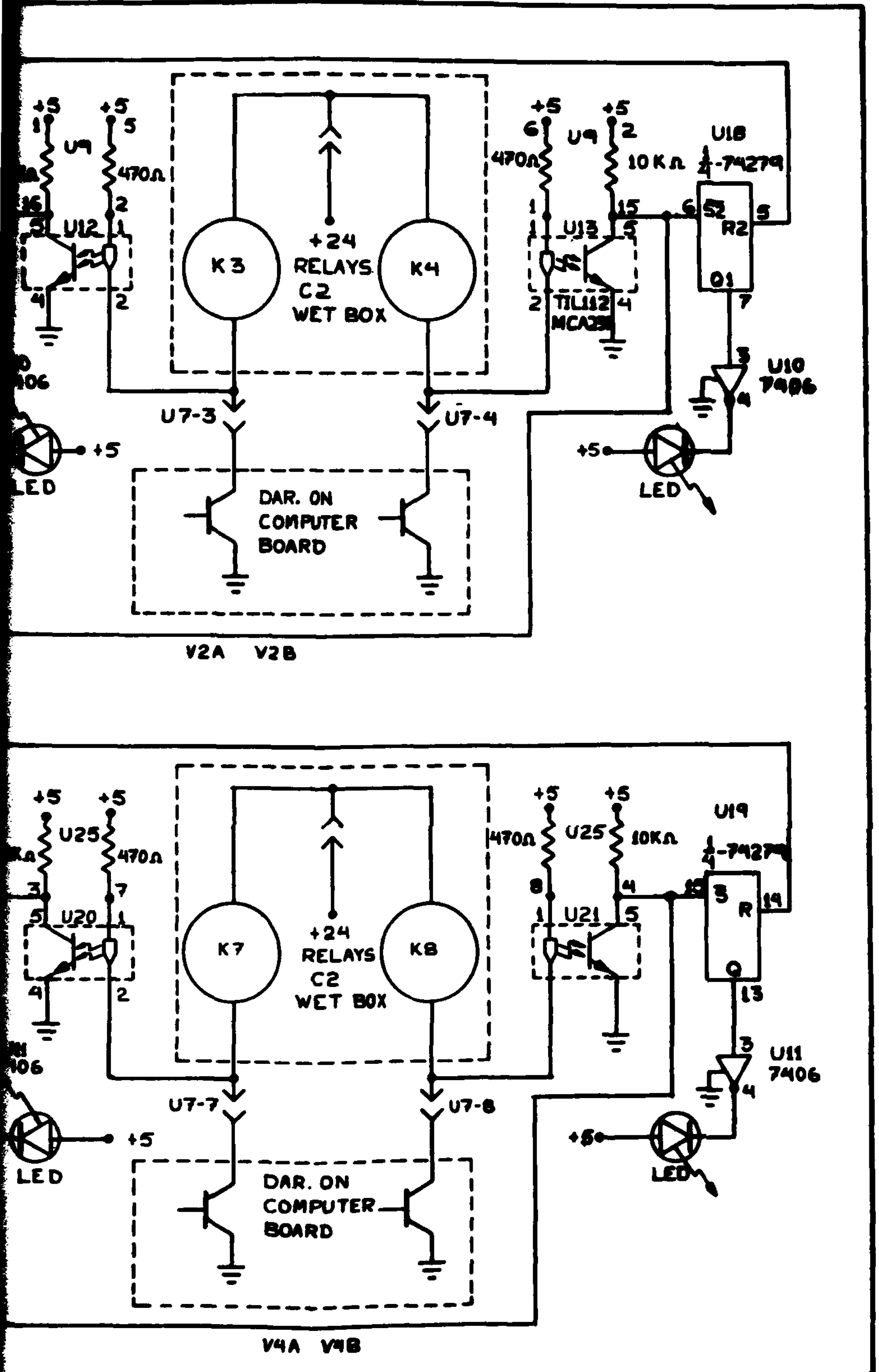
| DESIGN<br>C. CLAVELL | NOSC | DRAFT<br>JMF |
|---|---|---|
| DRAWING NO. CC-1 | 2-28-78 | REV. 3 |
| LOGIC DIAGRAM - SEAWATER ANAL.<br>TIMER INTERFACE TO<br>ELAPSED TIMER | | SHEET 1 OF 1 |



V1A - V1B

V3A V3B

74279



DESIGN
C. CLAVELL

NOSC

DRAFT
JMF.

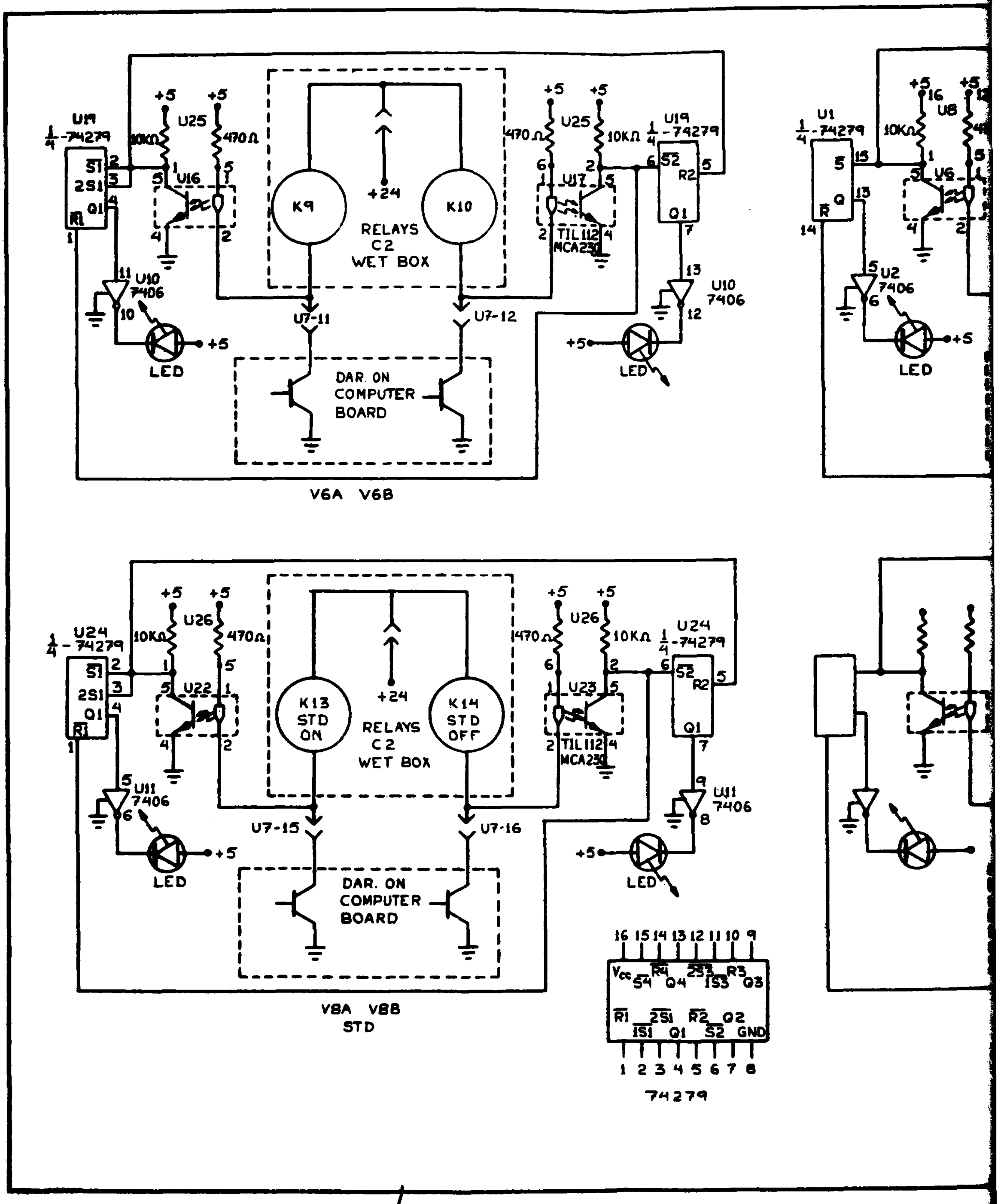
DRW# CC-2A

9-7-79

REV# 1

LED DISPLAY OF VALVE POSITION

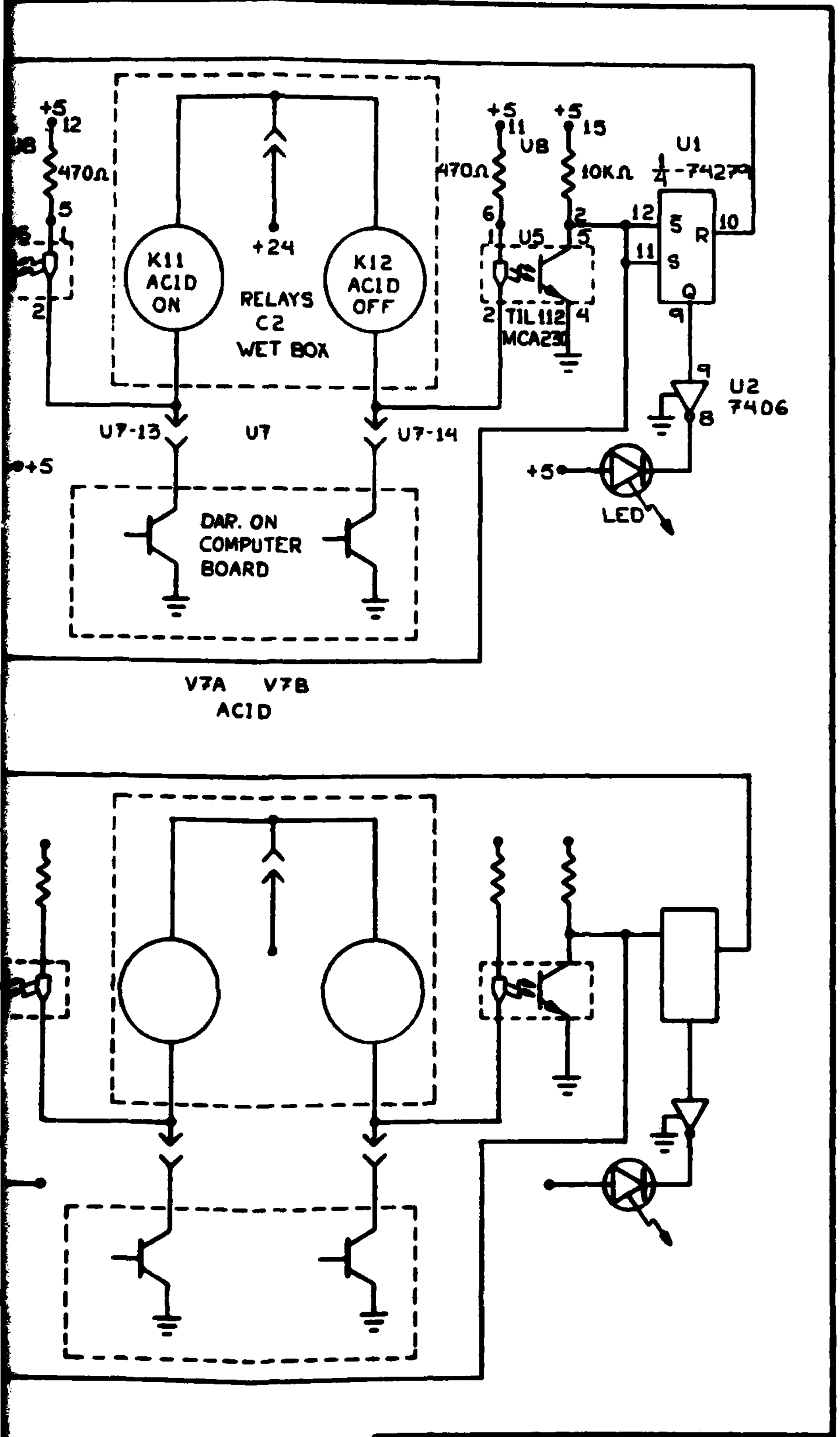
SHT 1 OF 2



V6A V6B

V8A V8B STD

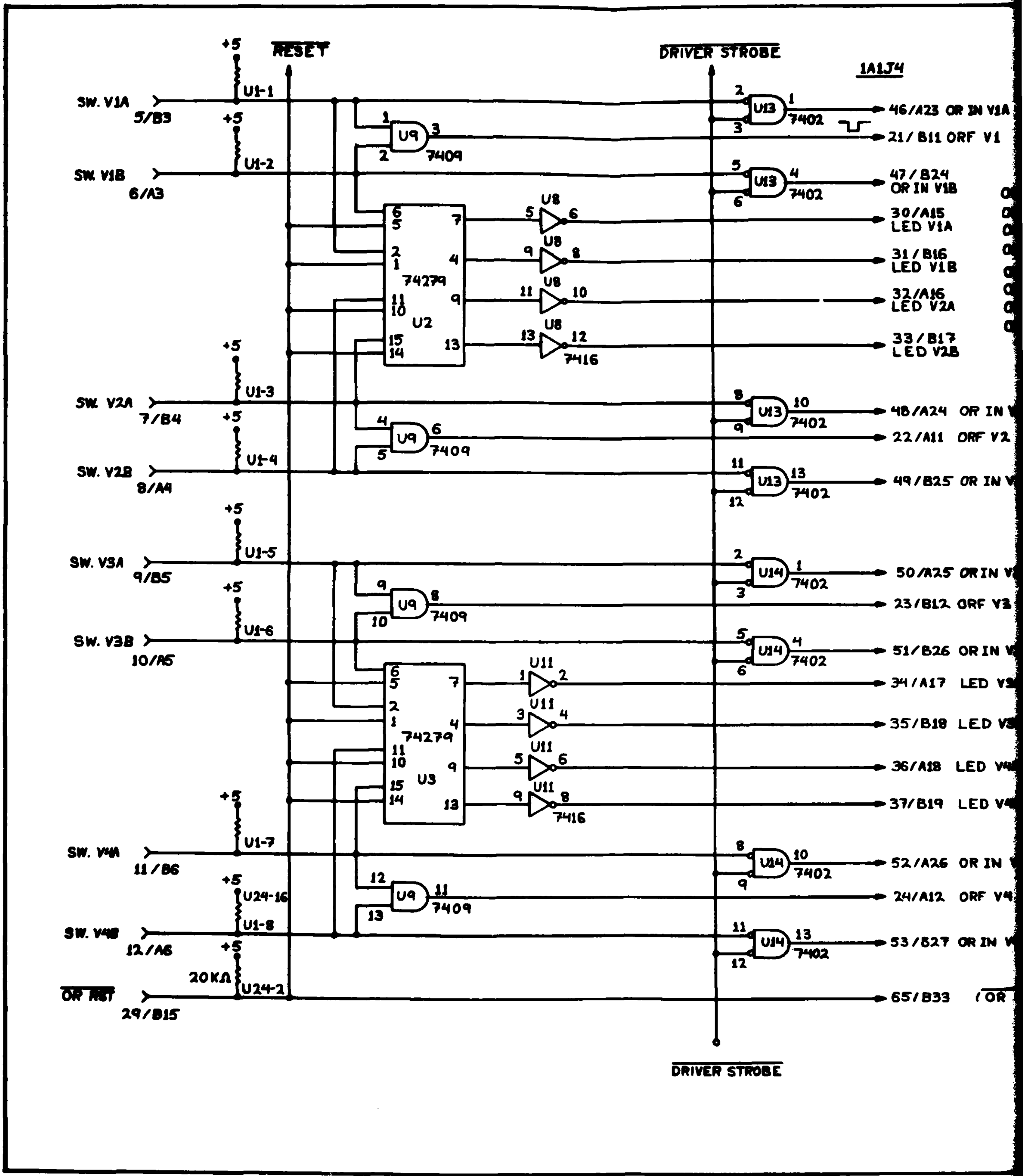
74279



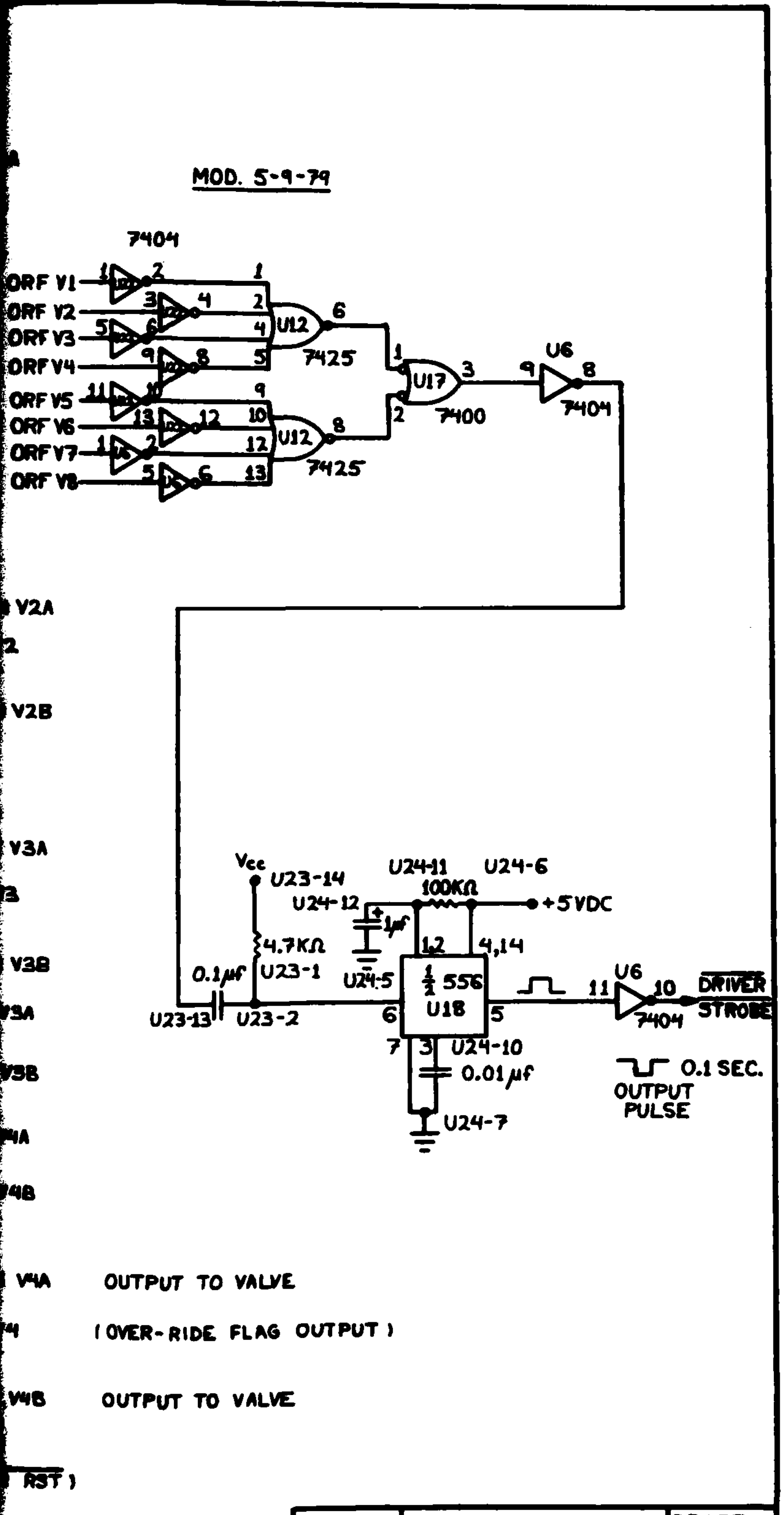
| DESIGN<br>C.CLAWELL | NOSC | DRAFT<br>JMF |
|---|---|---|
| DRW# CC-28 | 9-7-79 | REV# 1 |
| LED DISPLAY OF VALVE<br>POSITION | | SHT 2 OF 2 |

2

1


RESET
DRIVER STROBE
1A1J4
+5
SW. V1A
5/B3
U1-1
SW. V1B
6/A3
U1-2
U9
7409
U13
7402
46/A23 OR IN V1A
21/ B11 ORF V1
47/B24 OR IN V1B
74279
U2
U8
7416
30/A15 LED V1A
31/B16 LED V1B
32/A16 LED V2A
33/B17 LED V2B
SW. V2A
7/B4
U1-3
SW. V2B
8/A4
U1-4
48/A24 OR IN V
22/A11 ORF V2
49/B25 OR IN V
SW. V3A
9/B5
U1-5
SW. V3B
10/A5
U1-6
U14
50/A25 OR IN V
23/B12 ORF V3
51/B26 OR IN V
74279
U3
U11
34/A17 LED V3
35/B18 LED V3
36/A18 LED V4
37/B19 LED V4
SW. V4A
11/B6
U1-7
U24-16
SW. V4B
12/A6
U1-8
20KΩ
U24-2
OR RST
29/B15
52/A26 OR IN
24/A12 ORF V4
53/B27 OR IN V
65/B33
(OR

/

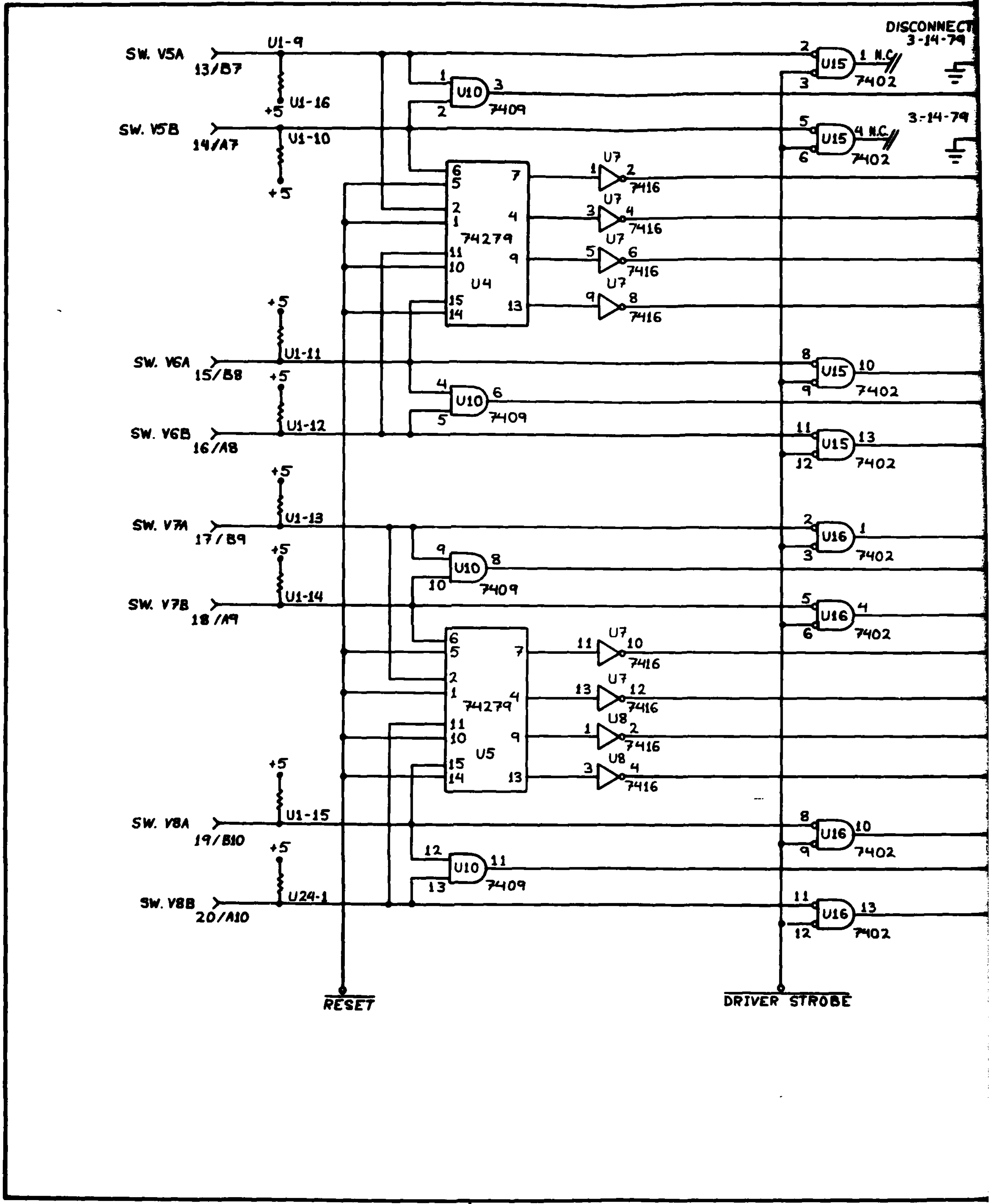


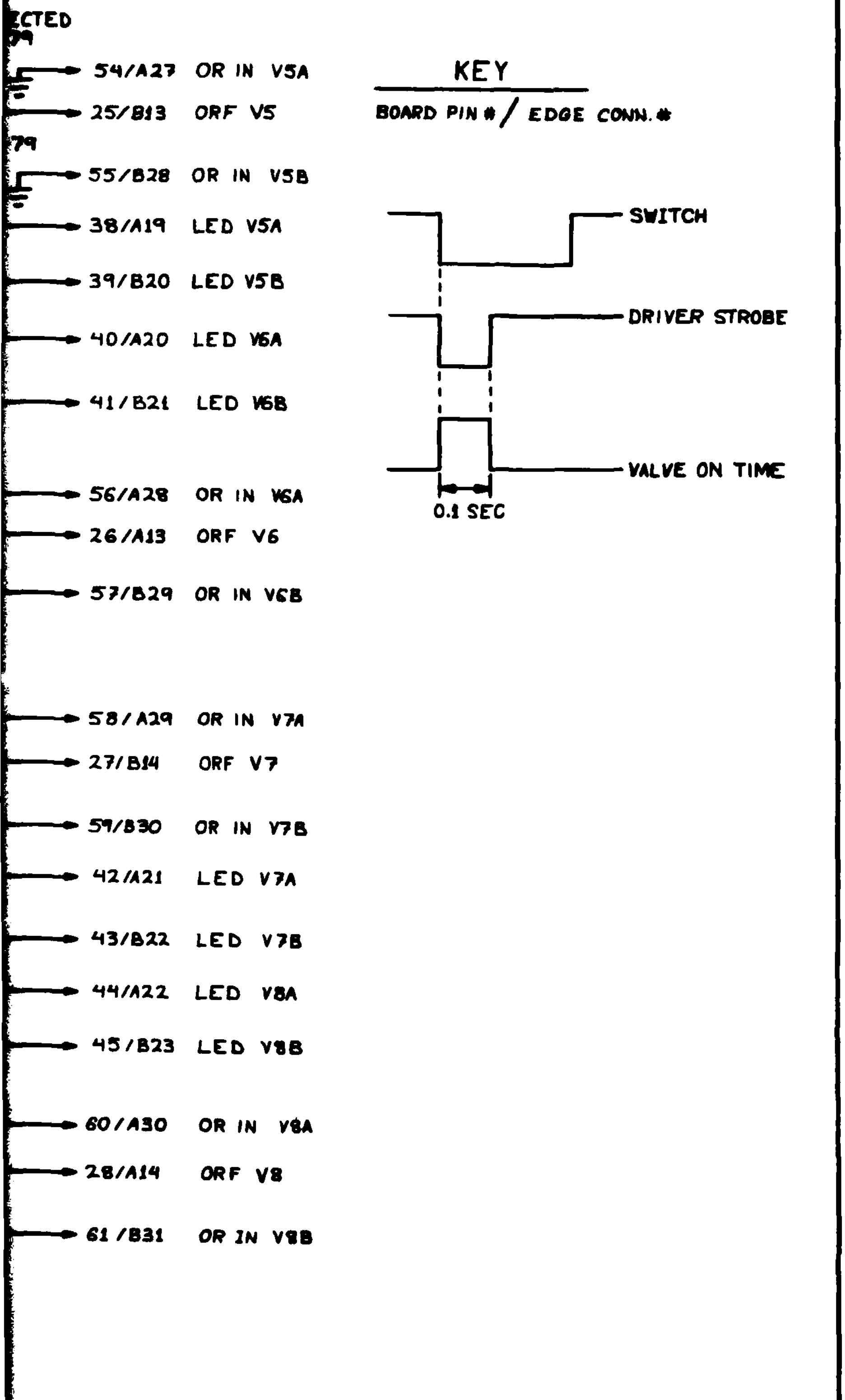
| DESIGN<br>C. CLAVELL | NOSC | DRAFT<br>JMF. |
|---|---|---|
| DRAWING NO. CC-2A | 5-9-79 | REV. 2 |

MANUAL VALVE OVERRIDE INTERFACE BOARD

SHEET 1 OF 2

2


DISCONNECT
3-14-79
3-14-79
SW. V5A
13/B7
U1-9
U1-16
+5
SW. V5B
14/A7
U1-10
+5
U10
7409
U15
7402
N.C.
U7
7416
74279
U4
SW. V6A
15/B8
U1-11
SW. V6B
16/A8
U1-12
SW. V7A
17/B9
U1-13
SW. V7B
18/A9
U1-14
U16
74279
U5
U8
SW. V8A
19/B10
U1-15
SW. V8B
20/A10
U24-1
RESET
DRIVER STROBE

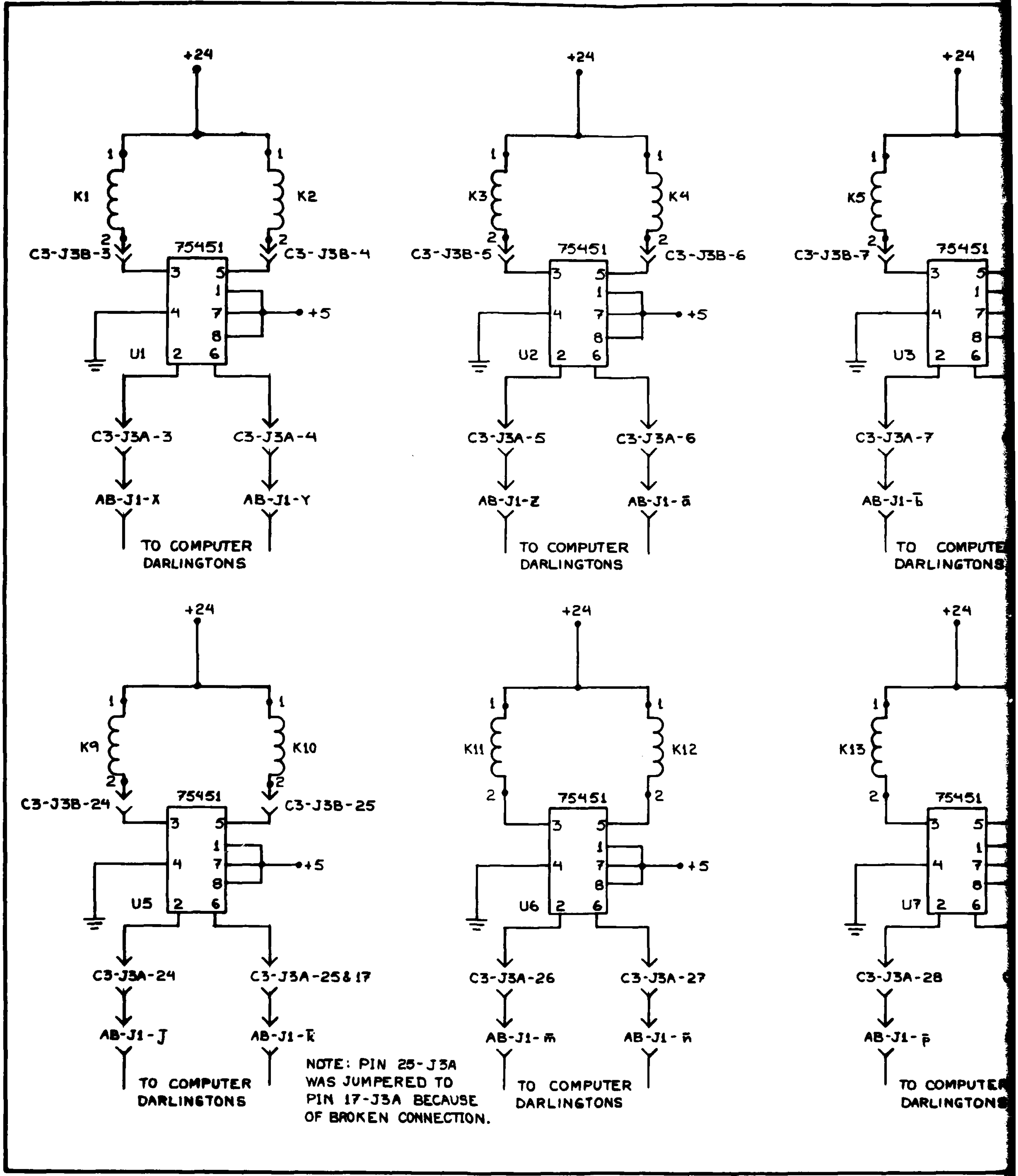


| DESIGN<br>C. CLAVELL | NOSC | DRAFT<br>JMF |
|---|---|---|
| DRAWING NO. CC-38 | 5-5-78 | REV. 2 |

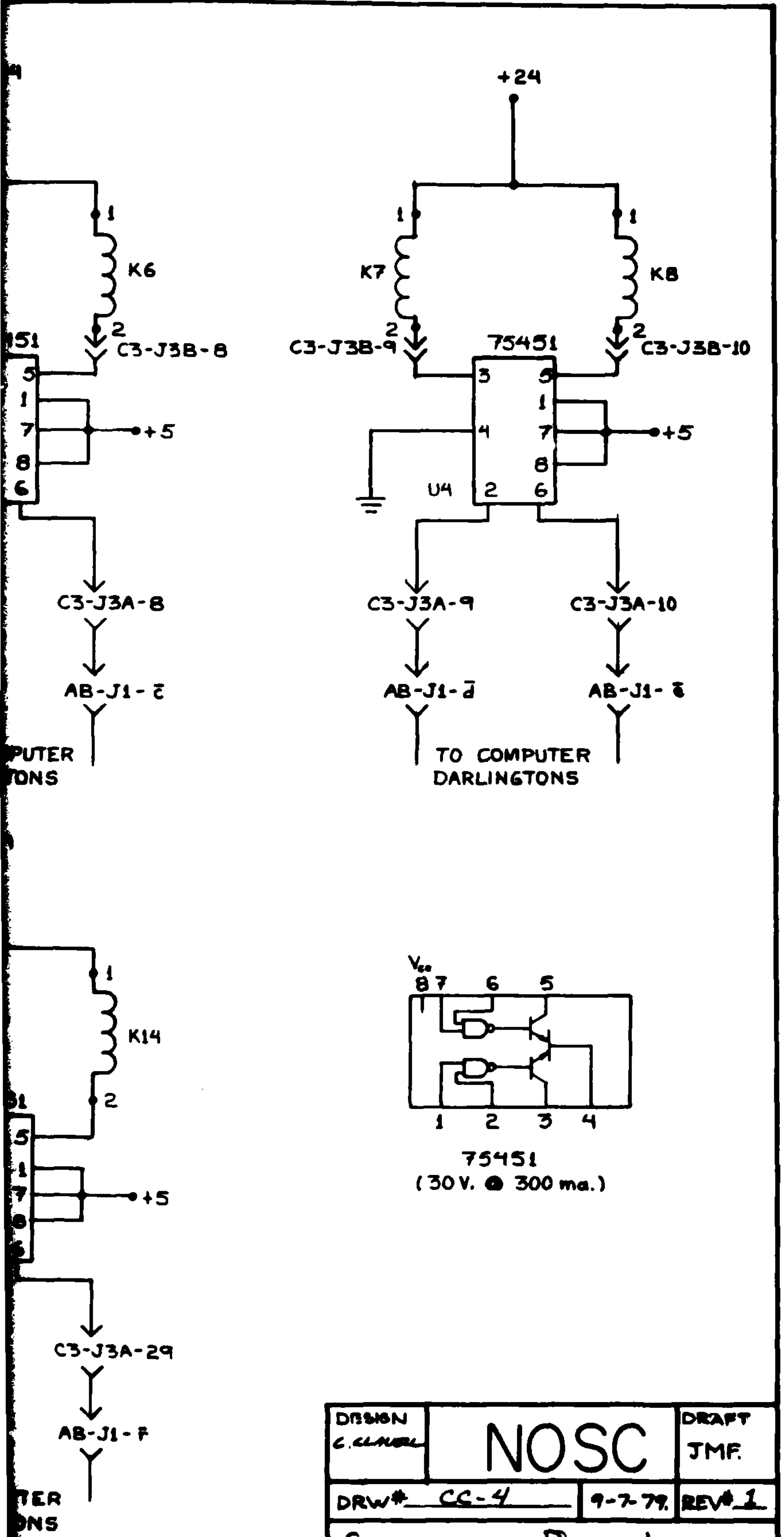
MANUAL VALVE OVERRIDE
INTERFACE BOARD

SHEET 2 OF 2

2



NOTE: PIN 25-J3A WAS JUMPERED TO PIN 17-J3A BECAUSE OF BROKEN CONNECTION.

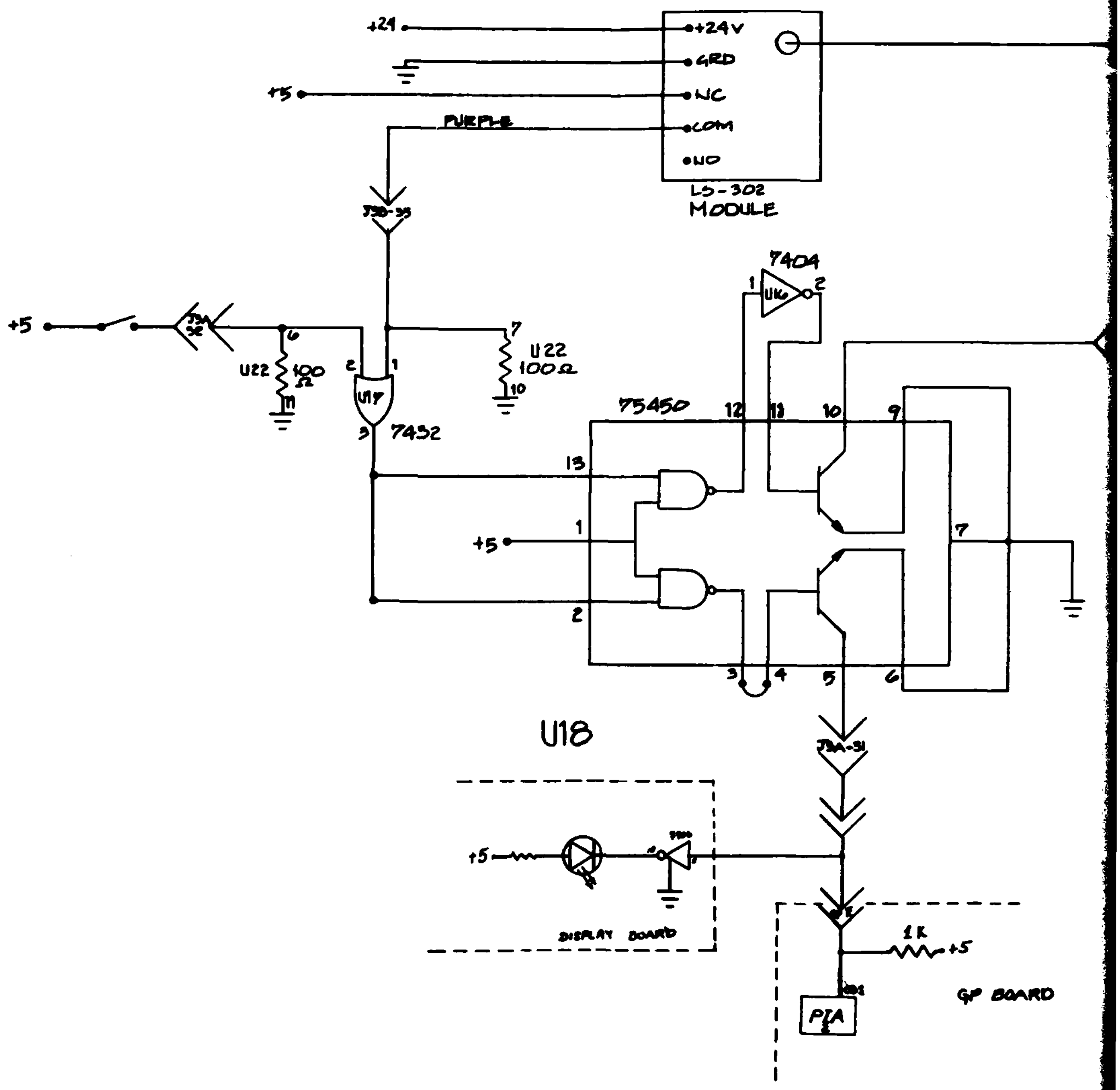


| DESIGN<br>C. [illegible] | NOSC | DRAFT<br>JMF. |
|---|---|---|
| DRW# CC-4 | 9-7-79. | REV# 1 |

COMPUTER TO RELAY INTERFACE

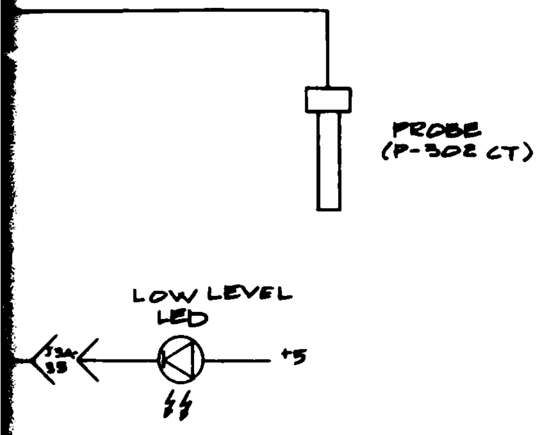
SHT 1 OF 1

ON LB-L3
& LS-302 MODULE
(LORION SCIENTIFC)



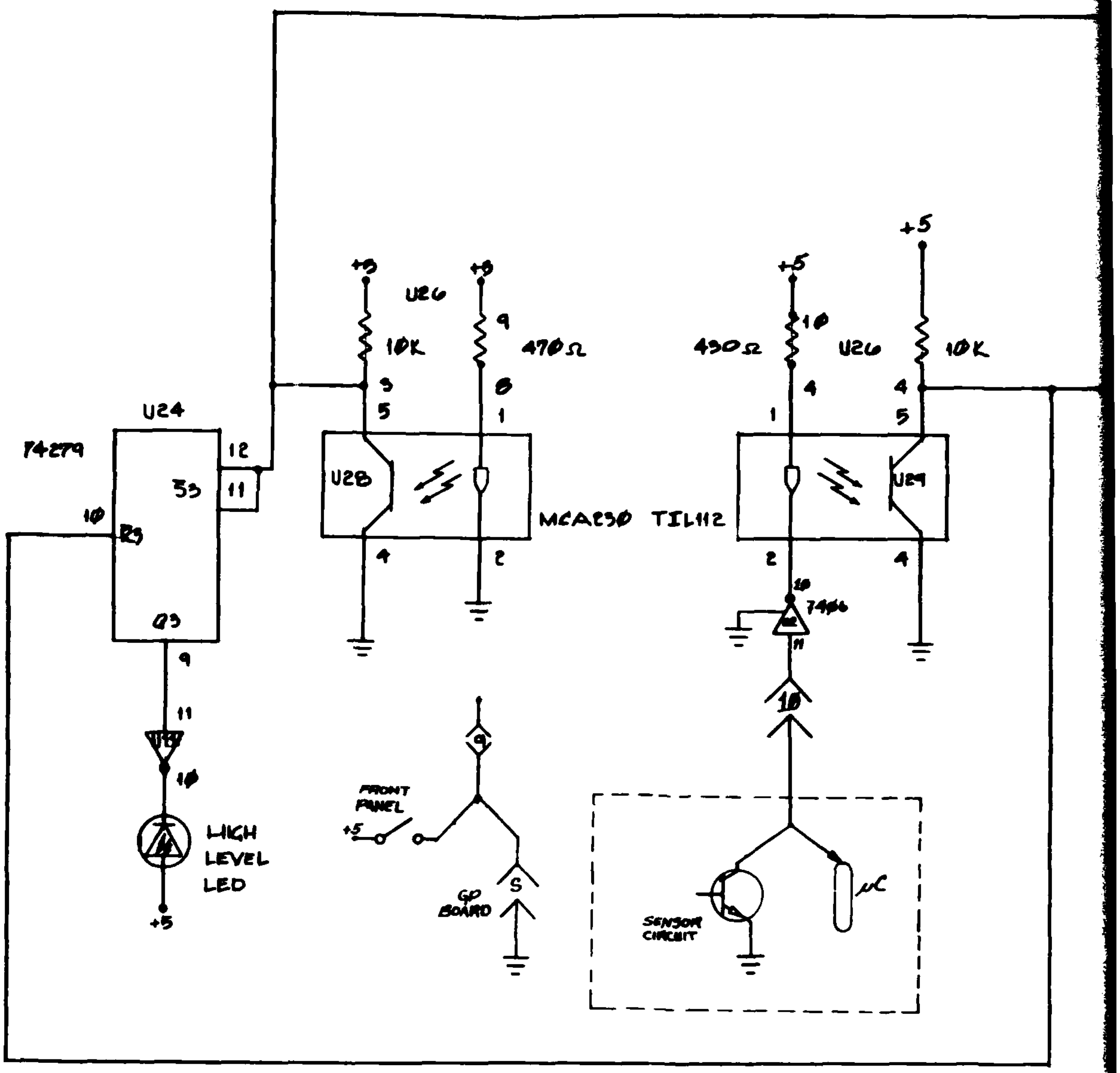
ACT. CONDITIONS

NC : OPEN - WET
NC : CLOSE - DRY

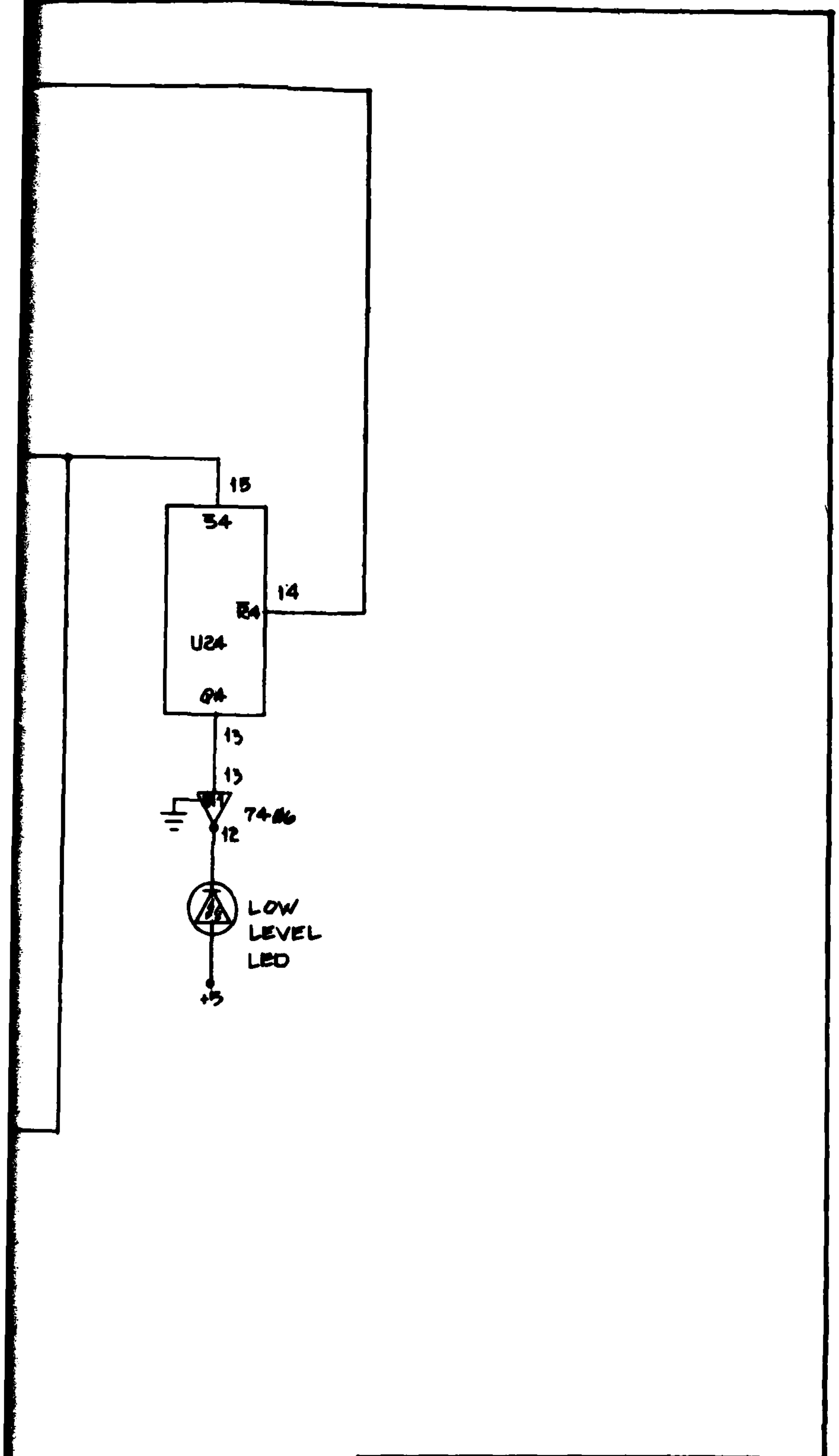


| | | |
|---|---|---|
| DESIGN<br>C. CLAVELL | NOSC | DRAFT<br>EMK |
| DRW# CC-5 | 7-9-78 | REV.# 1 |
| LOW LEVEL SENSOR CIRCUIT | | SHT 1 OF 1 |

2.

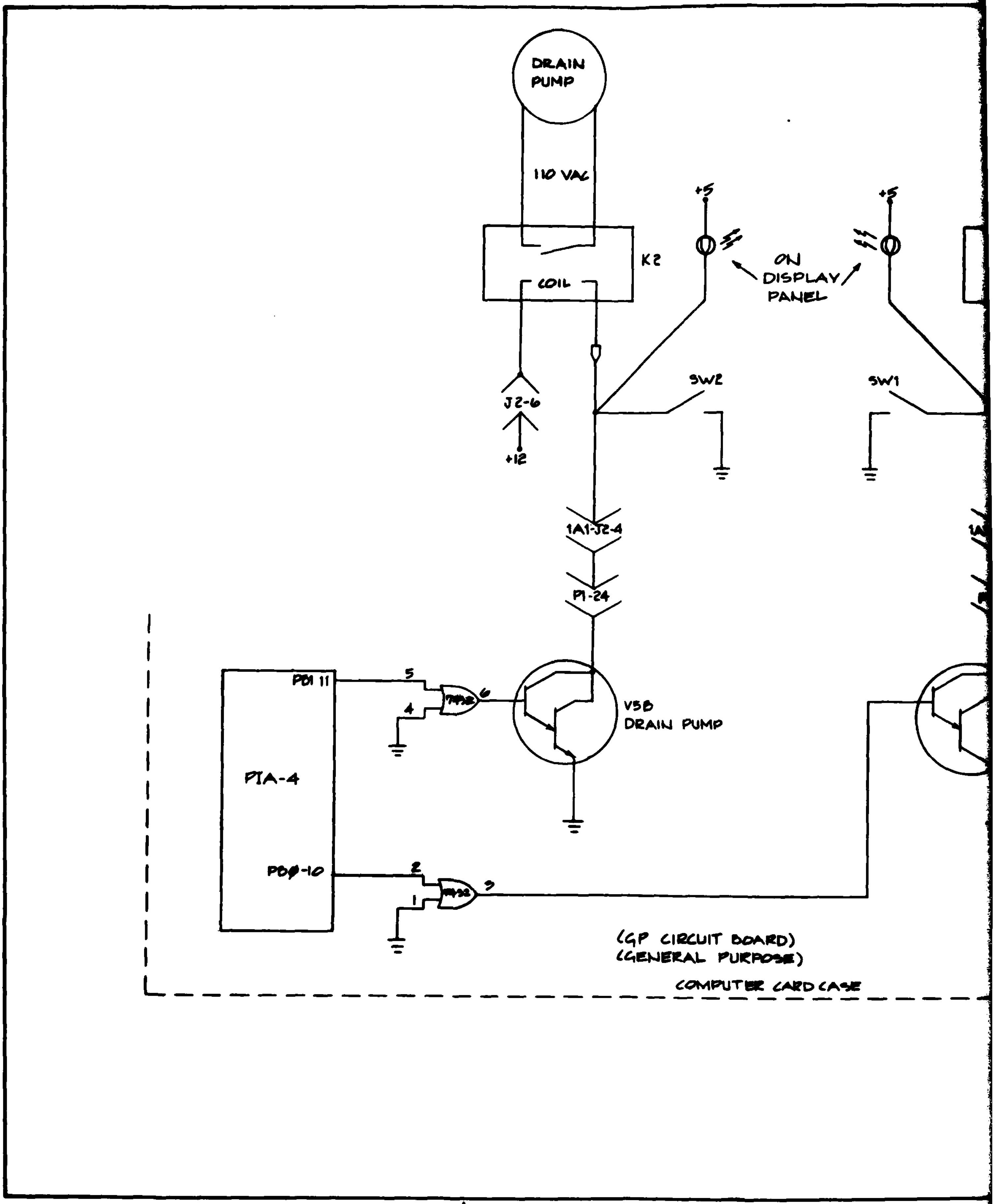


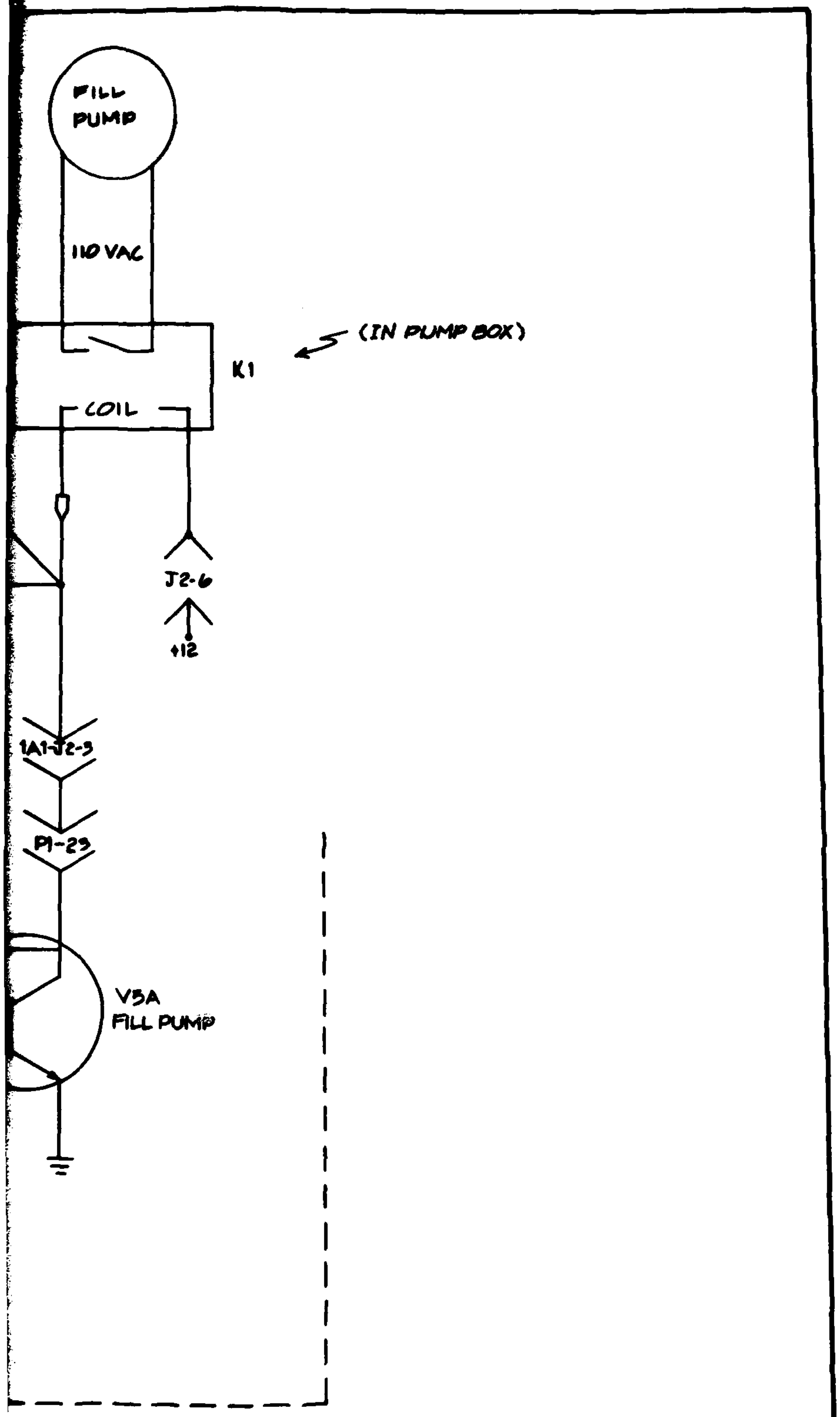
[ PART OF
PANEL DISPLAY BOARD ]



|  |  |  |
|---|---|---|
| DESIGN<br>C. CLAWELL | NOSC | DRAFT<br>EMK |
| DRW. # CC-6 | 9-7-79 | REV.# 1 |
| HIGH & LOW WATER LEVEL<br>SENSOR DISPLAY<br>CIRCUIT | SHT 1 OF 1 |  |

2


DRAIN PUMP
110 VAC
K2
COIL
+5
+5
ON DISPLAY PANEL
J2-6
+12
SW2
SW1
1A1-J2-4
P1-24
PB1 11
5
4
7432
6
V5B DRAIN PUMP
PIA-4
PBØ-10
2
1
7432
3
(GP CIRCUIT BOARD)
(GENERAL PURPOSE)
COMPUTER CARD CASE

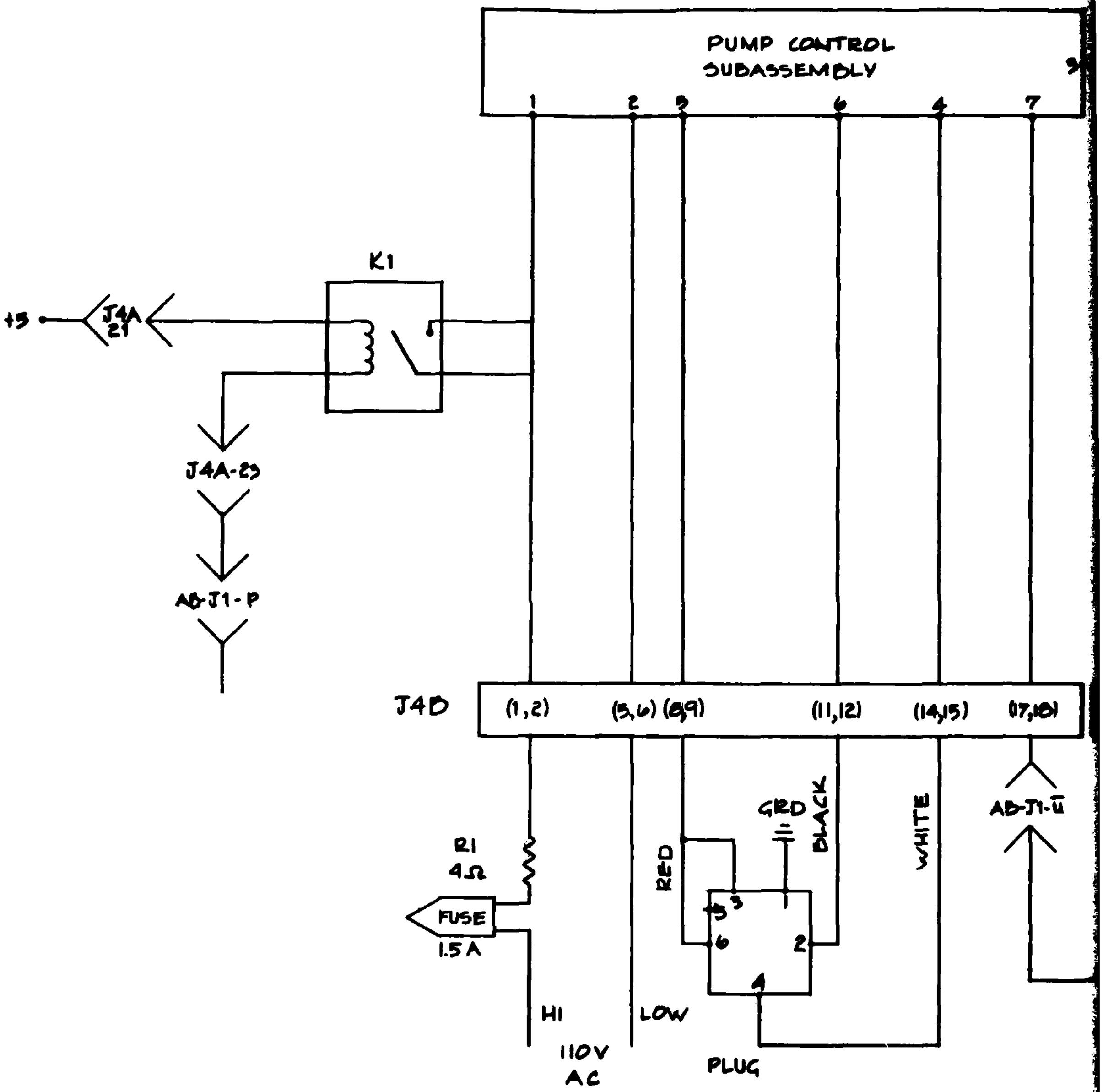


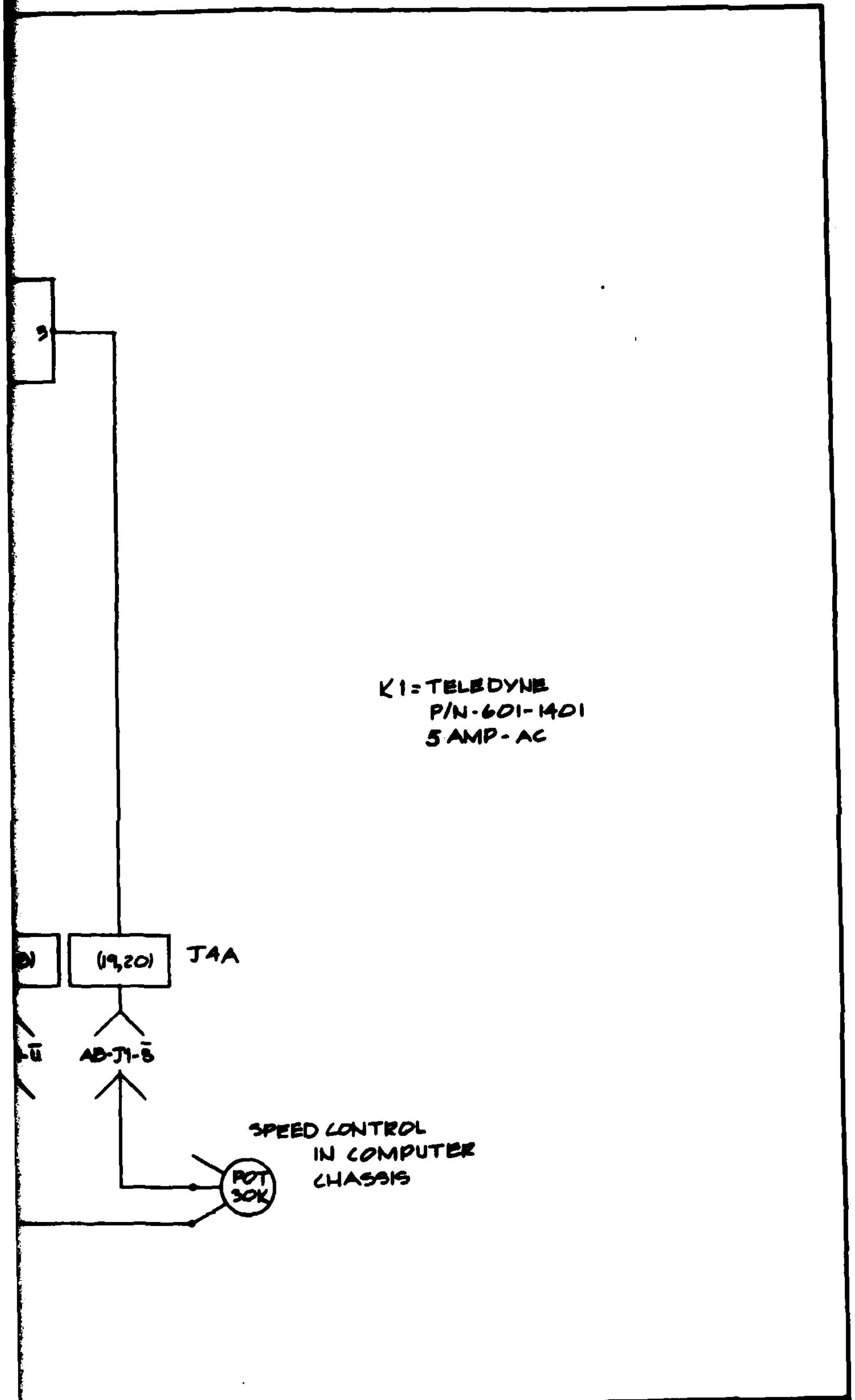
| | | |
|---|---|---|
| DESIGN<br>C.CLAWELL | NOSC | DRAFT<br>EMK |
| DRW# CC-7 | 9-7-79 | REV.# 1 |

Fill & Empty Pump Circuit

SHT 1 OF 1

MASTERFLEX PUMP CONTROLER



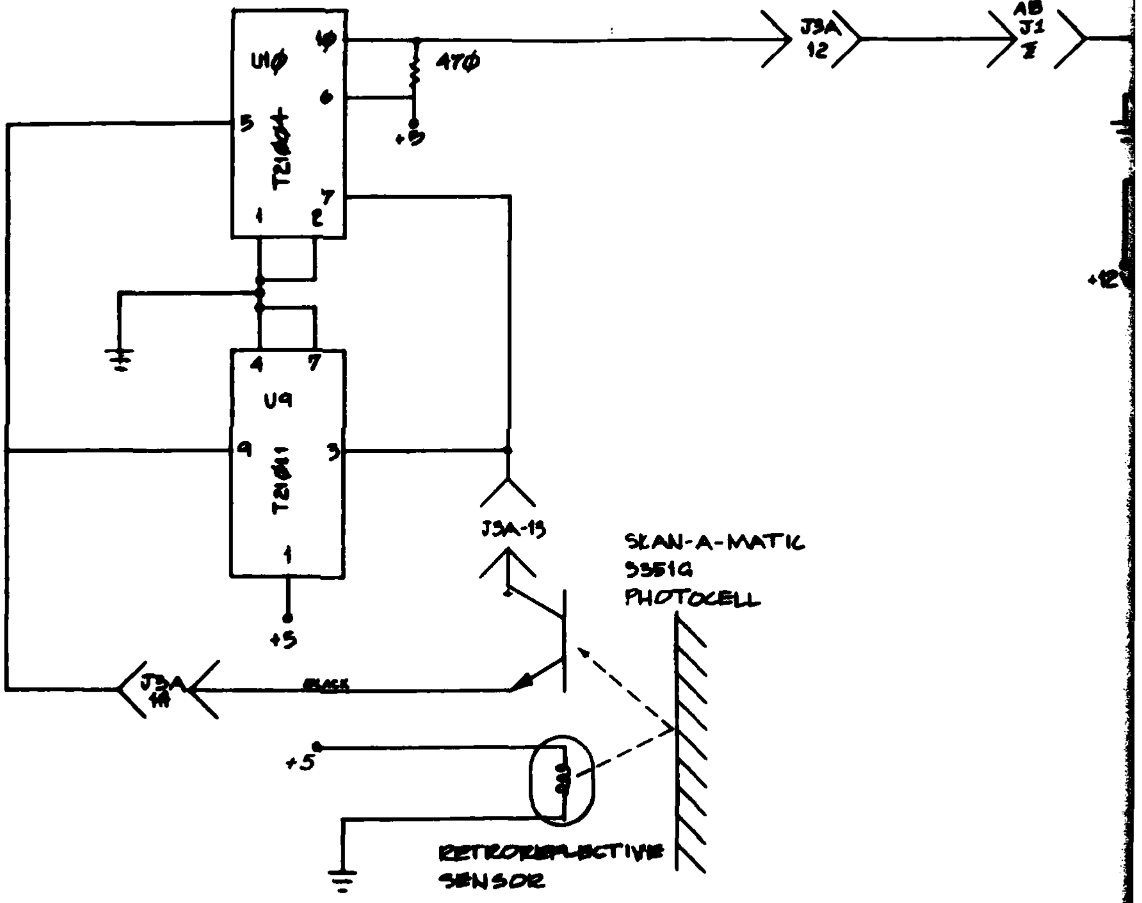


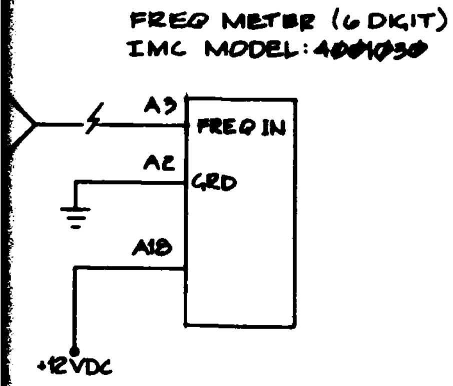
K1 = TELEDYNE
P/N-601-1401
5 AMP - AC

| | | |
|---|---|---|
| DESIGN<br>C.CLAWELL | NOSC | DRAFT<br>EMK. |
| DRW# CC-8 | 9-7-72 | REV# 1 |

PUMP CONTROLLER C4

SHT 1 OF 1


U1Ø
T21Ø4
1Ø
6
5
7
1
2
47Ø
+5
J3A
12
AB
J1
2
+12
4
7
U9
T21Ø1
9
3
1
+5
J3A-13
SKAN-A-MATIC
S351G
PHOTOCELL
J3A
14
BLACK
+5
RETROREFLECTIVE
SENSOR

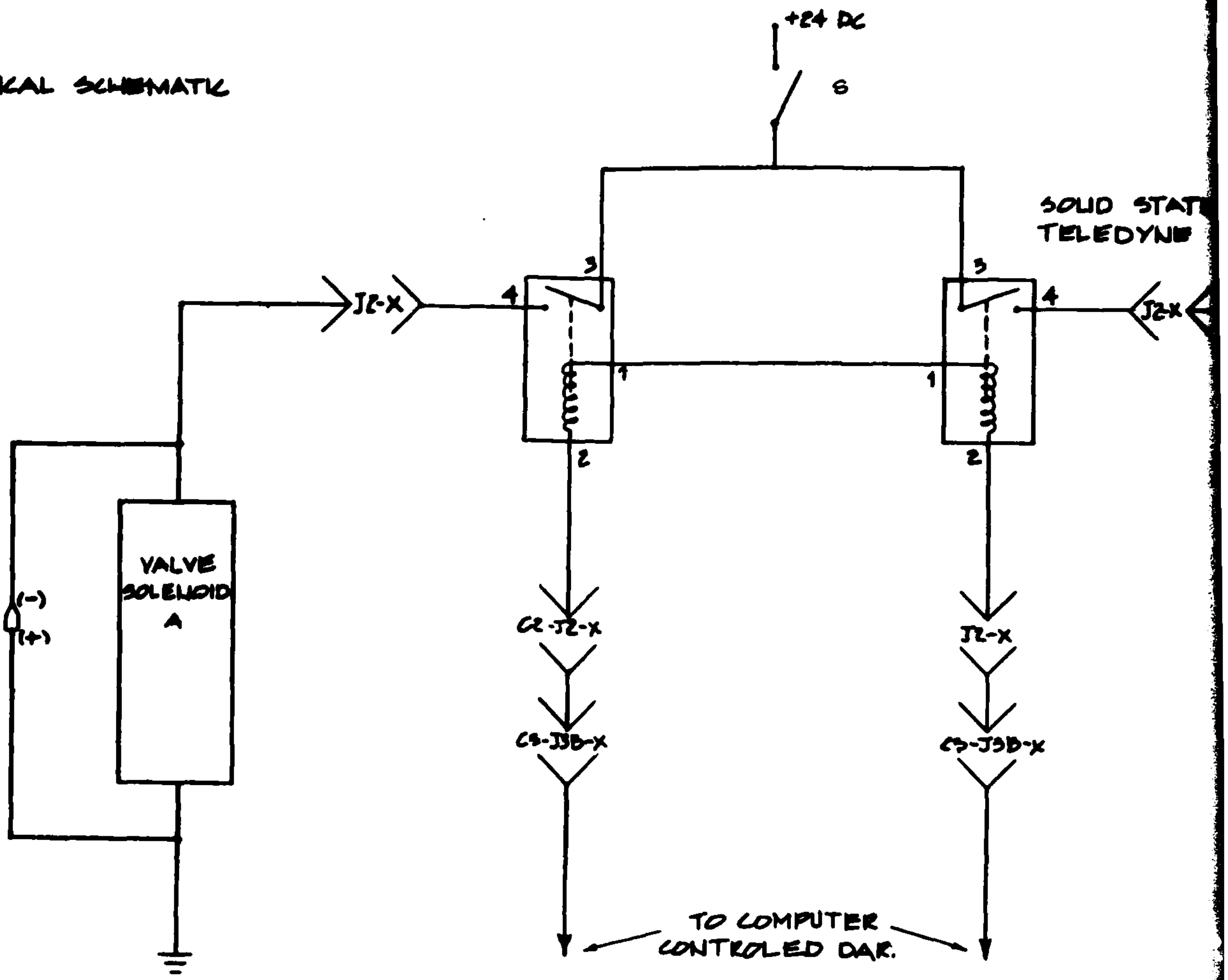


FREQ. METER
READS IN RPS

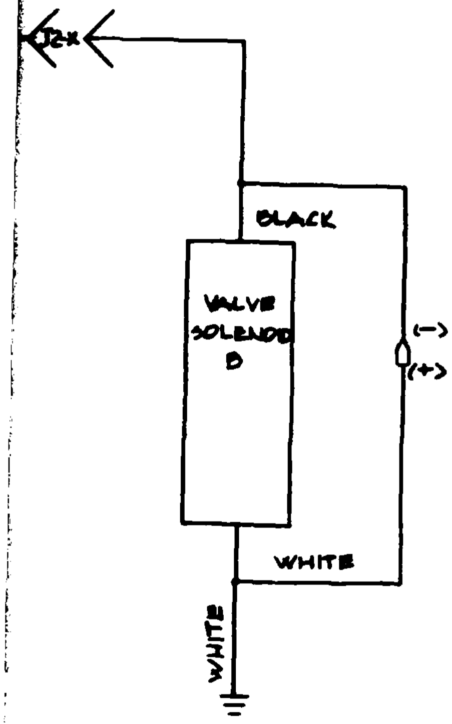
SKAN-A-MATIC CORP
MODULES & PHOTOCELL

| | | |
|---|---|---|
| DESIGN<br>CCLAVELL | NOSC | DRAFT<br>EMK. |
| DRW# CC-9 | 9-7-79 | REV# 1 |
| RPS SENSOR CIRCUIT<br>CARD 3 | | SHT 1 OF 1 |

TYPICAL SCHEMATIC



STATE RELAYS - 5AMP DC
YNE 6Ø3-2P

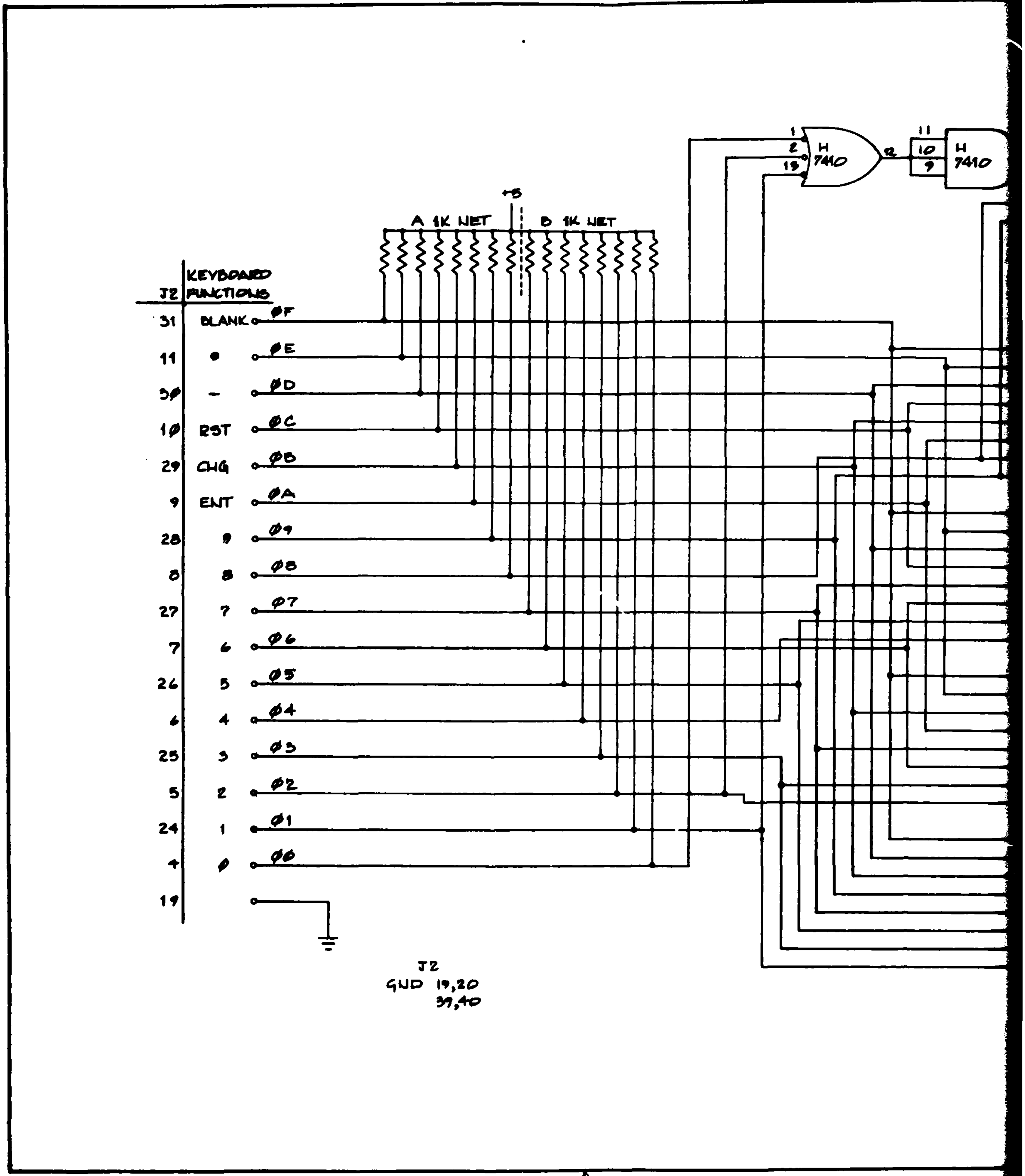


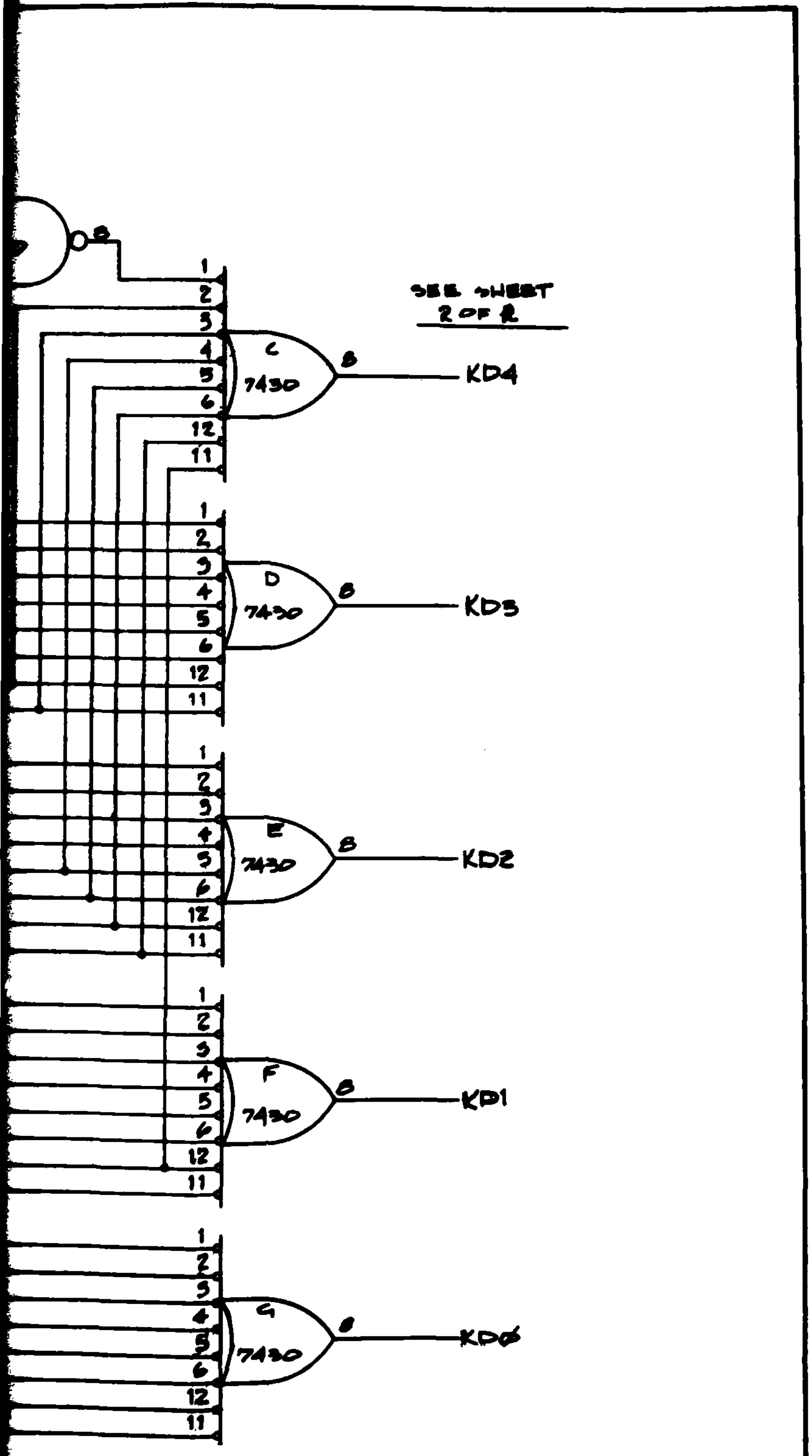
SOLENOIDS: DELTROL
24 VDC - PULSE
#53623-83

| | | |
|---|---|---|
| DESIGN<br>C LLANEZ | NOSC | DRAFT<br>EMK |
| DRW # CC-1Ø | 9-7-79 | REV# 1 |

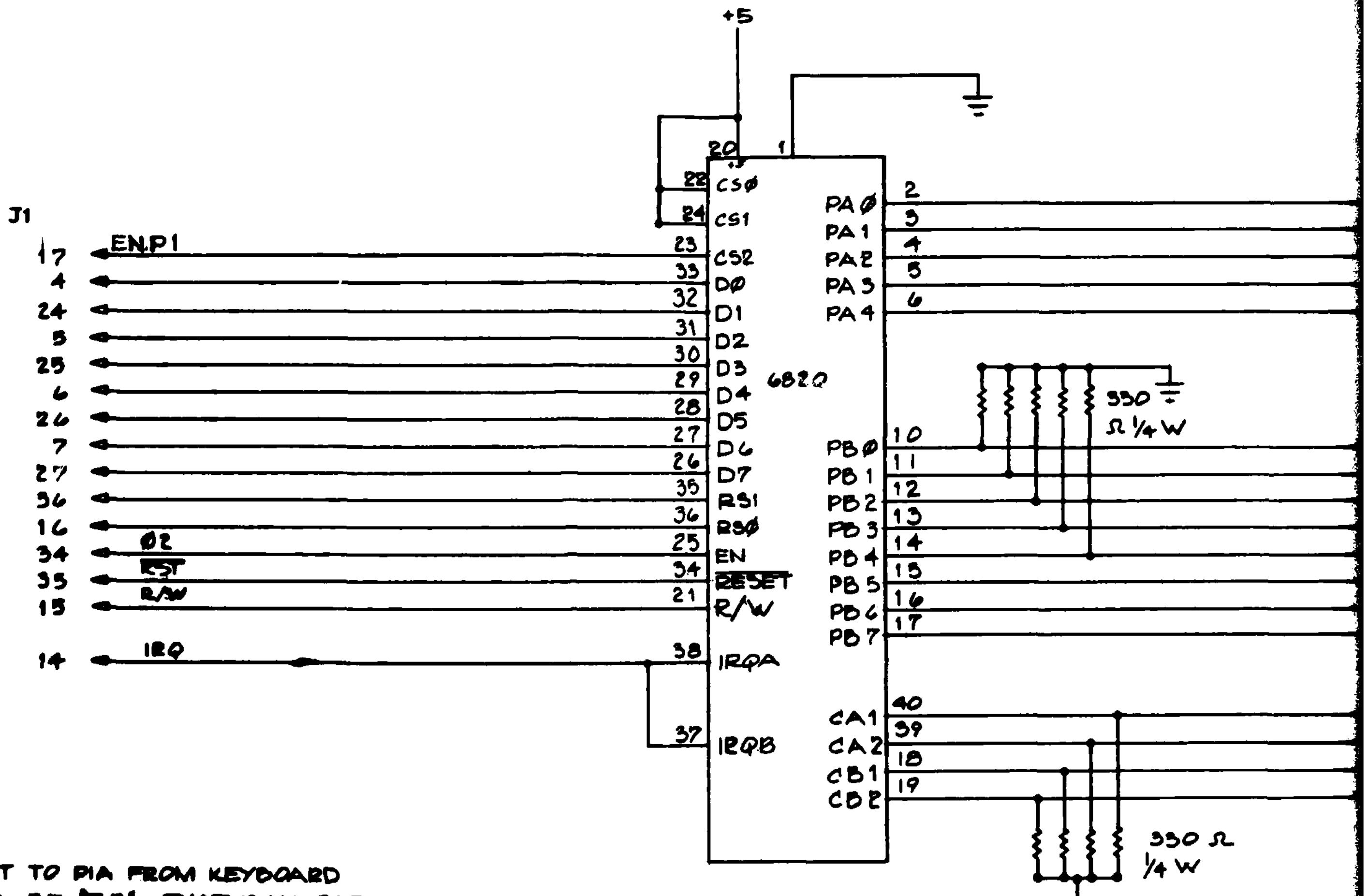
MAIN VALVE BOARD RELAY
BOARD CARD 2

SHT 1 OF 1


J2
KEYBOARD FUNCTIONS
31 BLANK ØF
11 • ØE
3Ø – ØD
1Ø RST ØC
29 CHG ØB
9 ENT ØA
28 9 Ø9
8 8 Ø8
27 7 Ø7
7 6 Ø6
26 5 Ø5
6 4 Ø4
25 3 Ø3
5 2 Ø2
24 1 Ø1
4 Ø ØØ
19
+5
A 1K NET
B 1K NET
1
2
13
H
7410
12
11
10
9
H
7410
J2
GND 19,20
39,40



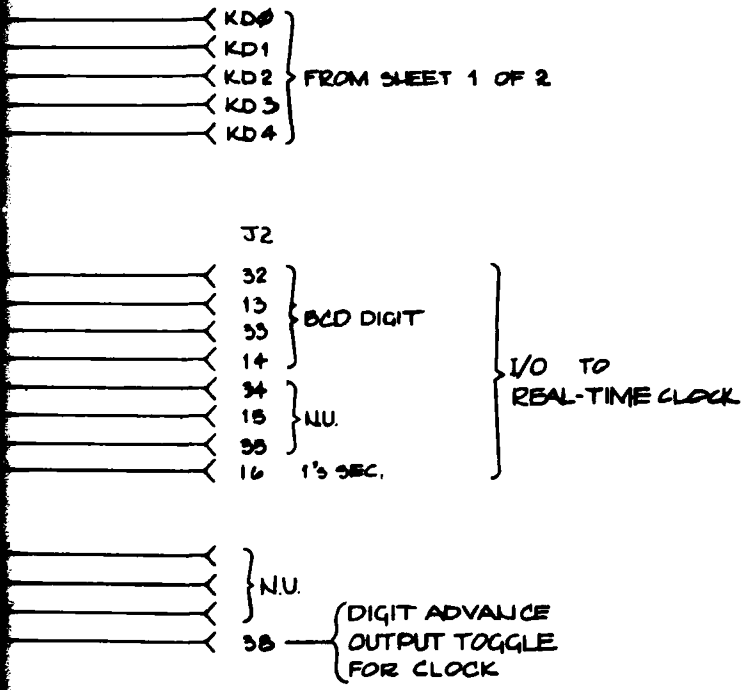
| | | |
|---|---|---|
| DESIGN<br>C.CLAVELL | NOSC | DRAFT<br>EMK |
| DRW# CC-11A | 9-7-79 | REV# 1 |
| SEA WATER ANALYZER<br>EXTRA BOARD 1 | | SHT 1 OF 2 |



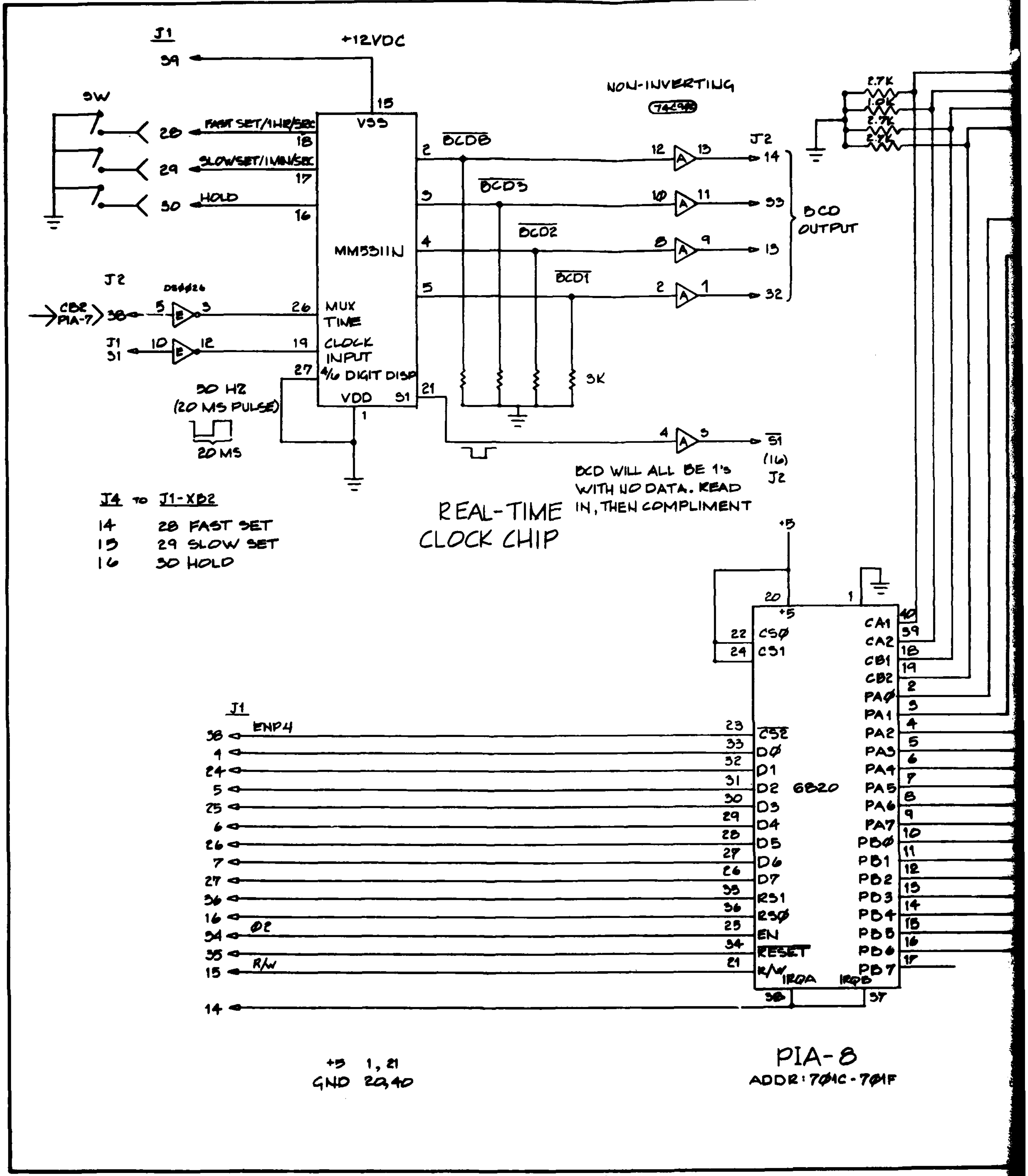
PIA 7
ADDR: 7Ø1Ø-7Ø13

INPUT TO PIA FROM KEYBOARD
WILL BE 'EØ' - THAT IS ALL BUT
THE UPPER 3 BITS WILL BE Ø IF
NO KEY IS HELD DOWN

J1
+5 1, 21
GND 20,40



| | | |
|---|---|---|
| DESIGN<br>CLLAHELL | NOSC | DRAFT<br>EMK |
| DRW# CC-11B | 9-7-77. | REV# 1 |
| SEA WATER ANALYZER<br>EXTRA BOARD 1 | | SHT 2 OF 2 |



BCD WILL ALL BE 1's WITH NO DATA. READ IN, THEN COMPLIMENT

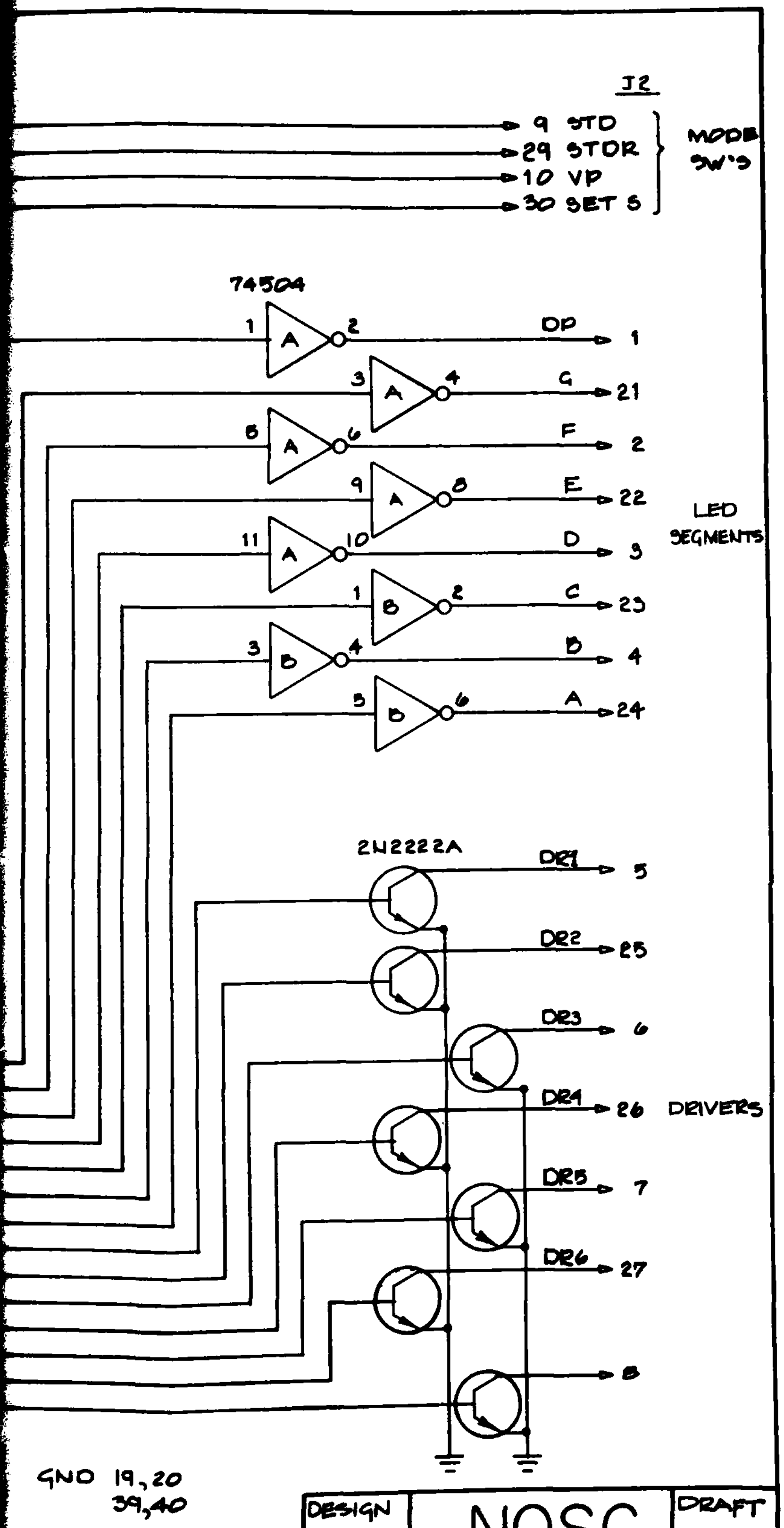
J4 to J1-XB2

| J4 | J1-XB2 |
|---|---|
| 14 | 28 FAST SET |
| 15 | 29 SLOW SET |
| 16 | 30 HOLD |

REAL-TIME CLOCK CHIP

+5 1, 21
GND 20,40

PIA-8
ADDR: 701C-701F



DESIGN

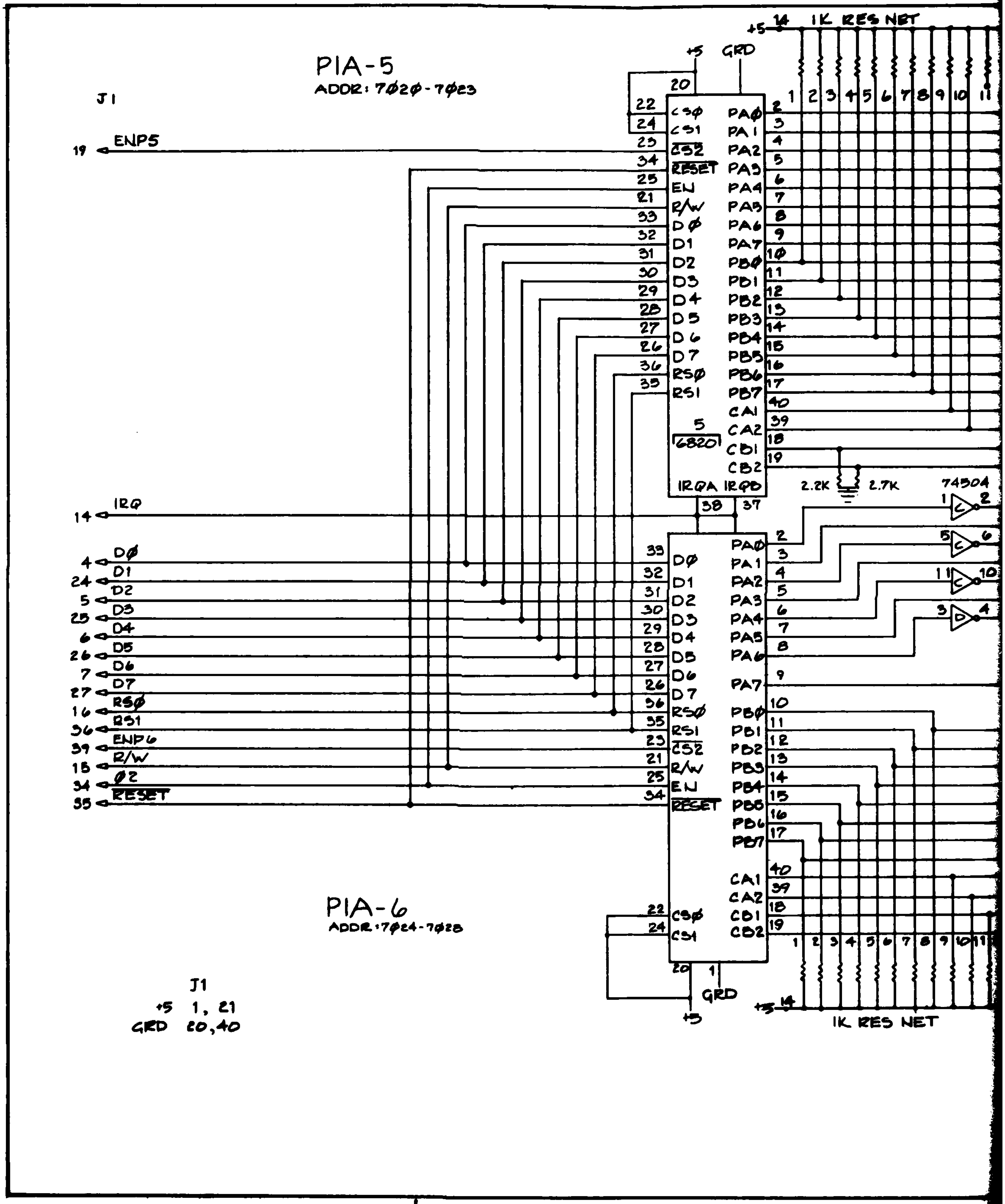
NOSC

DRAFT EMK

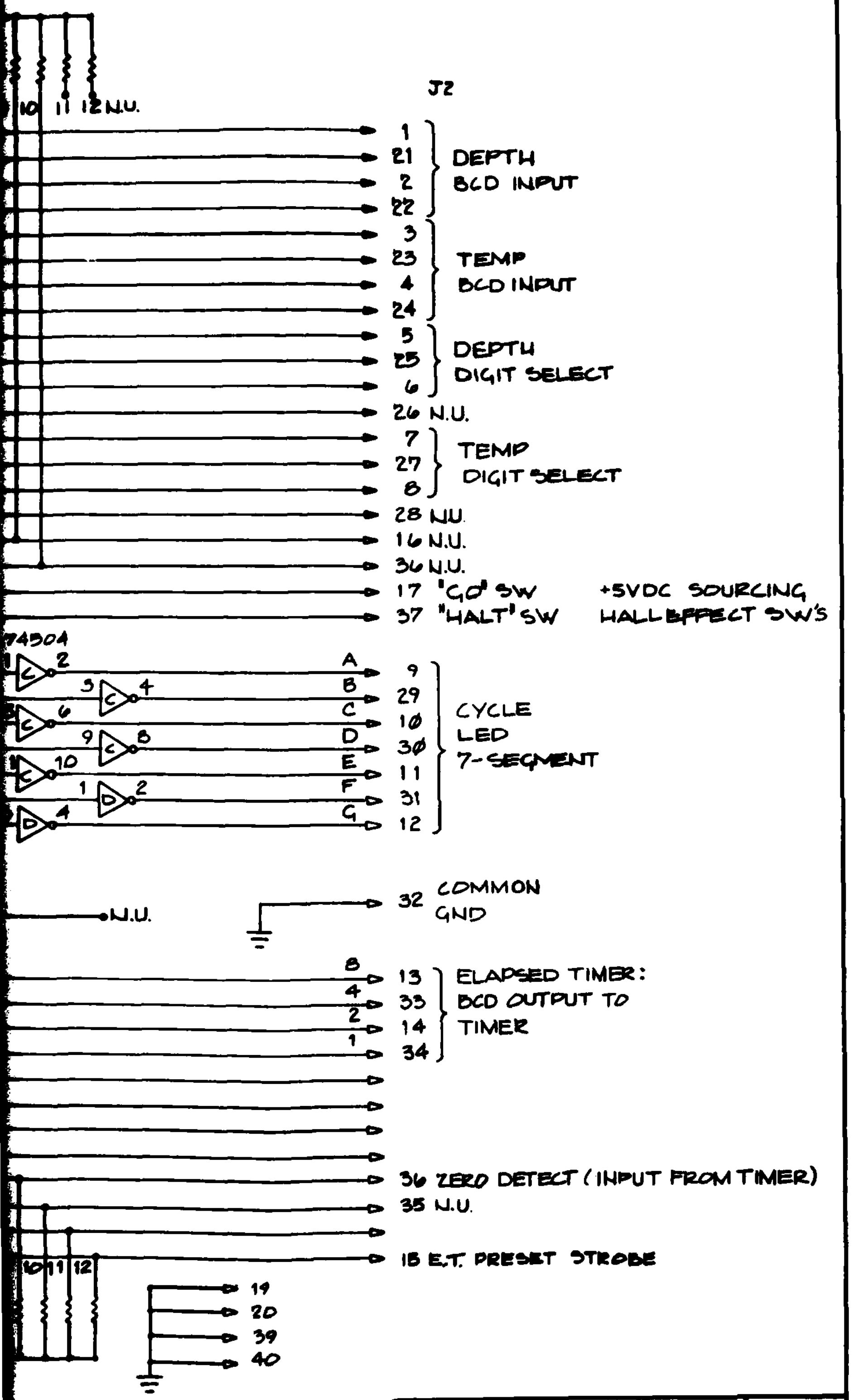
DRW# CC-12 9-7-79. REV# 1

SEA WATER ANALYZER
EXTRA BOARD 2

SHT 1 OF 1

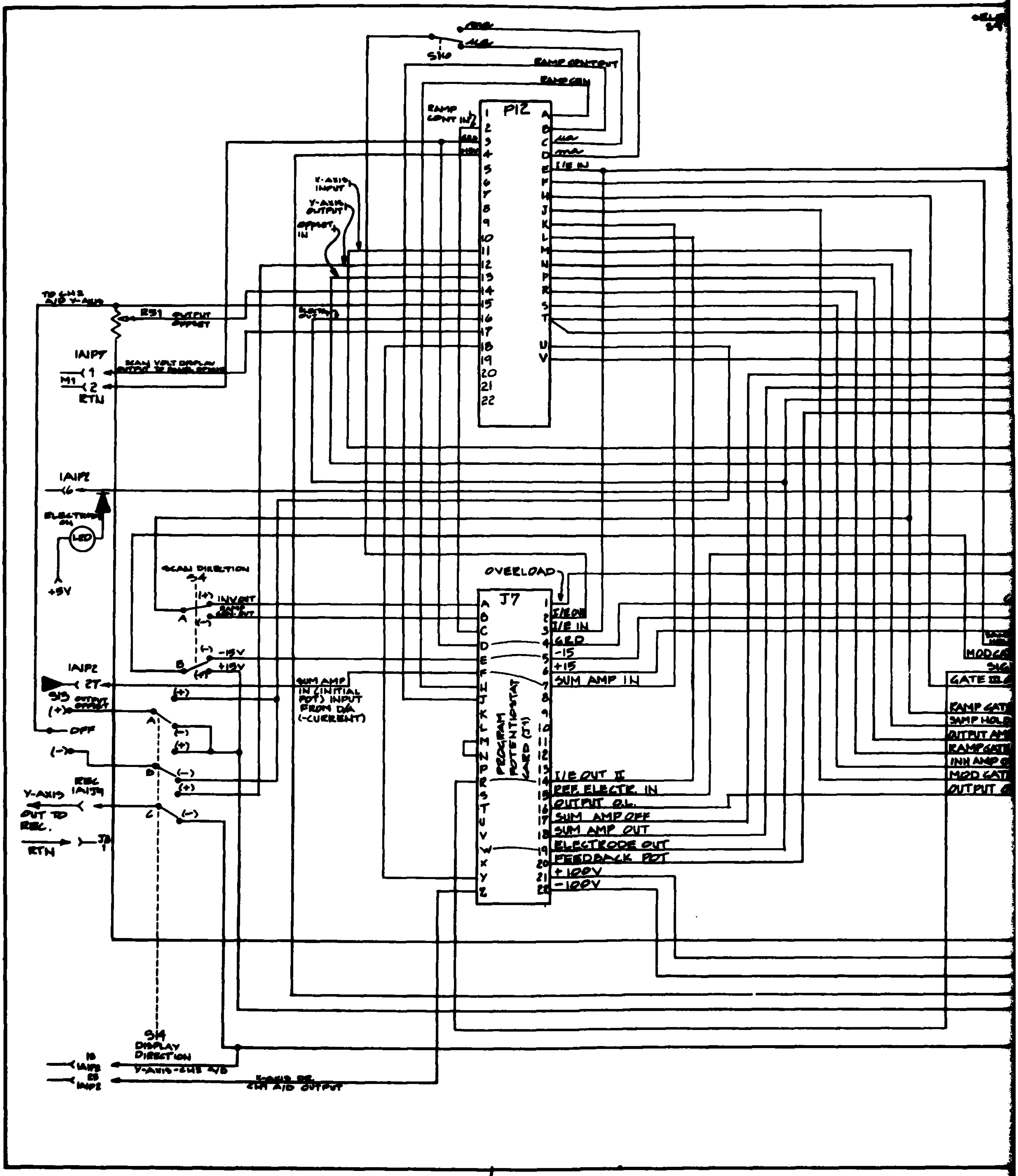


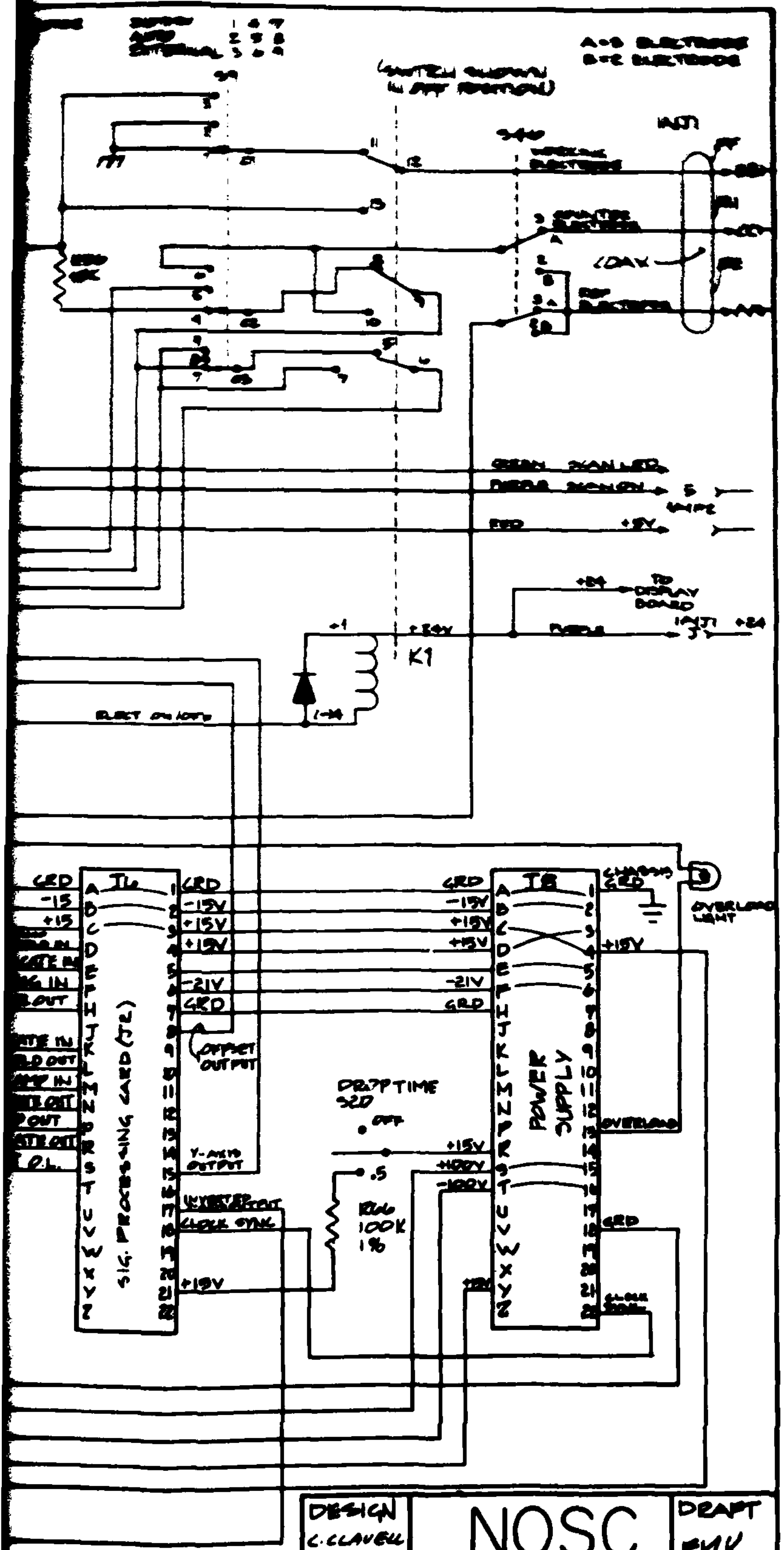
PIA-5
ADDR: 7Ø2Ø-7Ø23
J1
19 ENP5
14 IRQ
4 DØ
24 D1
5 D2
25 D3
6 D4
26 D5
7 D6
27 D7
16 RSØ
36 RS1
39 ENP6
15 R/W
34 Ø2
35 RESET
+5
GRD
20
22 CSØ
24 CS1
23 CS2
34 RESET
25 EN
21 R/W
33 DØ
32 D1
31 D2
30 D3
29 D4
28 D5
27 D6
26 D7
36 RSØ
35 RS1
PAØ 2
PA1 3
PA2 4
PA3 5
PA4 6
PA5 7
PA6 8
PA7 9
PBØ 1Ø
PB1 11
PB2 12
PB3 13
PB4 14
PB5 15
PB6 16
PB7 17
CA1 40
CA2 39
CB1 18
CB2 19
5
6820
IRQA IRQB
38 37
+5 14 1K RES NET
1 2 3 4 5 6 7 8 9 10 11
2.2K
2.7K
74S04
1 2
5 6
11 10
3 4
PAØ 2
PA1 3
PA2 4
PA3 5
PA4 6
PA5 7
PA6 8
PA7 9
PBØ 10
PB1 11
PB2 12
PB3 13
PB4 14
PB5 15
PB6 16
PB7 17
CA1 40
CA2 39
CB1 18
CB2 19
33 DØ
32 D1
31 D2
30 D3
29 D4
28 D5
27 D6
26 D7
36 RSØ
35 RS1
23 CS2
21 R/W
25 EN
34 RESET
22 CSØ
24 CS1
20
1
GRD
+5
PIA-6
ADDR: 7Ø24-7Ø28
1 2 3 4 5 6 7 8 9 10 11
+5 14
1K RES NET
J1
+5 1, 21
GRD 20,40



| DESIGN<br>C.CLAVELL | NOSC | DRAFT<br>EMK. |
|---|---|---|
| DRW# CC-13 | 9-7-79 | REV# 1 |

SEA WATER ANALYZER
EXTRA BOARD 3 SHT 1 OF 1

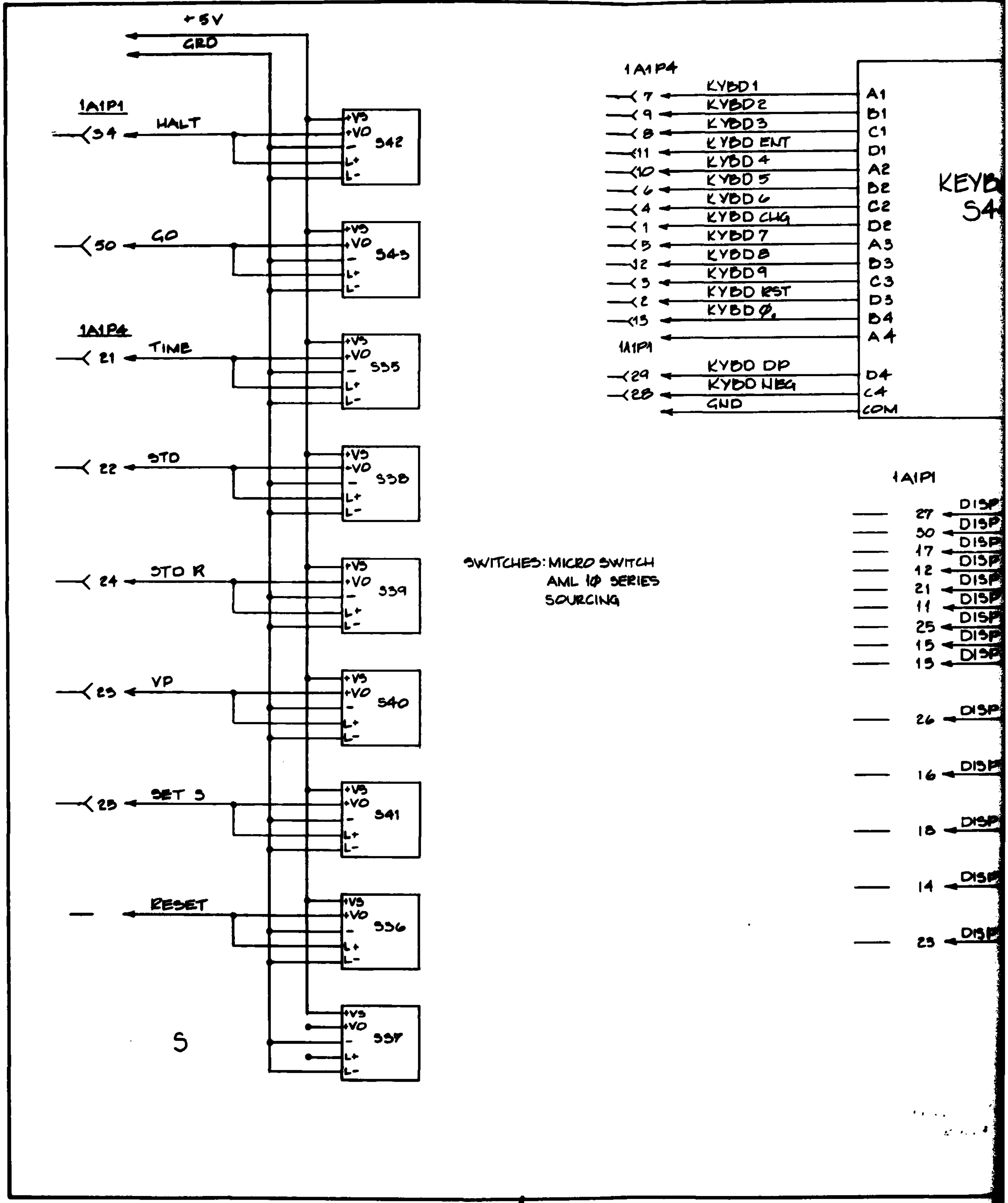

S10
RAMP CONT OUT
RAMP GEN
RAMP CONT IN
P12
I/E IN
X-AXIS INPUT
Y-AXIS OUTPUT
OFFSET IN
TO CH2 A/D Y-AXIS
R51
OUTPUT OFFSET
ELECTR OUT
1A1P7
SCAN VOLT DISPLAY OUTPUT TO PANEL METER
M1
RTN
1A1P2
ELECTRODE ON
LED
+5V
SCAN DIRECTION
S4
INVERT
RAMP GEN OUT
-15V
+15V
OVERLOAD
J7
I/E OUT
I/E IN
GRD
-15
+15
SUM AMP IN
SUM AMP IN (INITIAL POT) INPUT FROM D/A (-CURRENT)
PROGRAM POTENTIOSTAT CARD (J7)
S13
OUTPUT OFFSET
OFF
REC 1A1J4
Y-AXIS OUT TO REC.
RTN
J3
I/E OUT II
REF. ELECTR. IN
OUTPUT O.L.
SUM AMP OFF
SUM AMP OUT
ELECTRODE OUT
FEEDBACK POT
+100V
-100V
S14
DISPLAY DIRECTION
Y-AXIS-CH3 A/D
X-AXIS OR CH1 A/D OUTPUT
GATE III
RAMP GATE
SAMP HOLD
OUTPUT AMP
RAMP GATE
INH AMP
MOD GATE
OUTPUT



| DESIGN<br>C. CLAVELL | NOSC | DRAFT<br>EMK |
|---|---|---|
| DRW# CC-14 | 9-7-79 | REV# 1 |

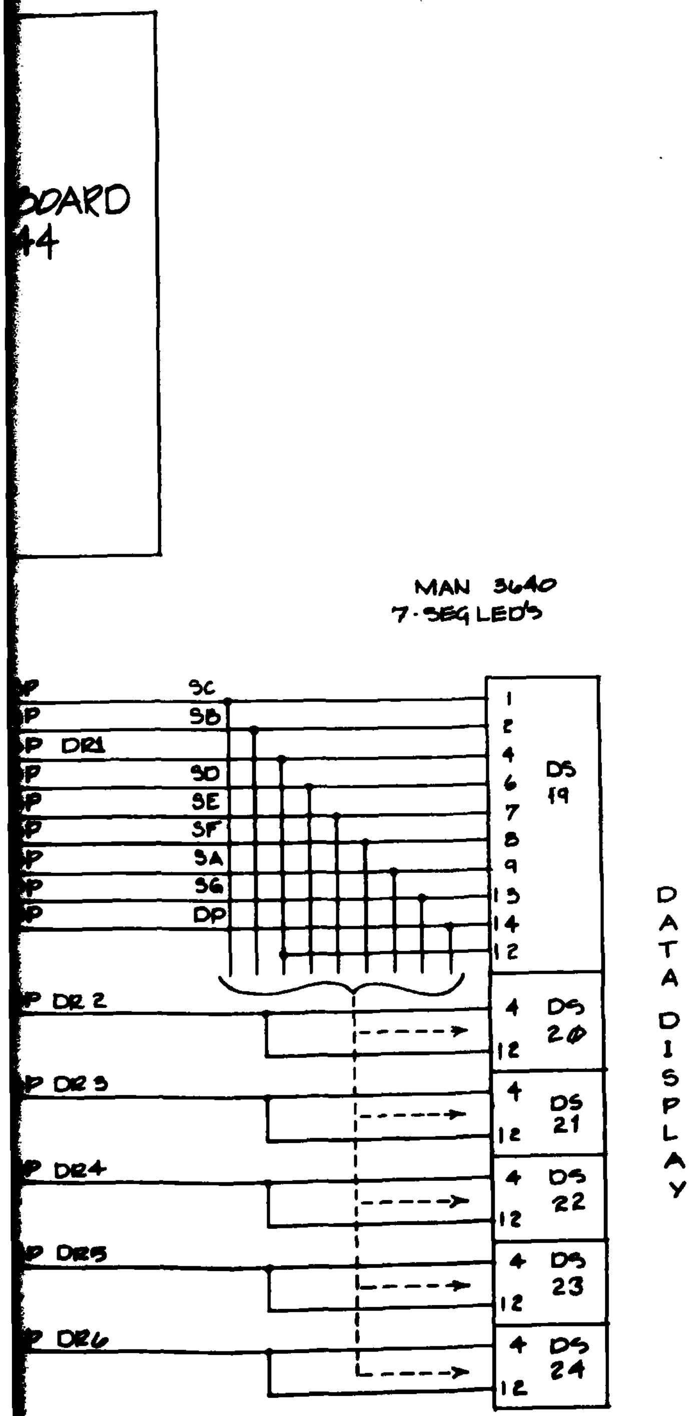
POTENTIOSTAT BACKPLANE

SHT 1 OF 1



SWITCHES: MICRO SWITCH
AML 10 SERIES
SOURCING

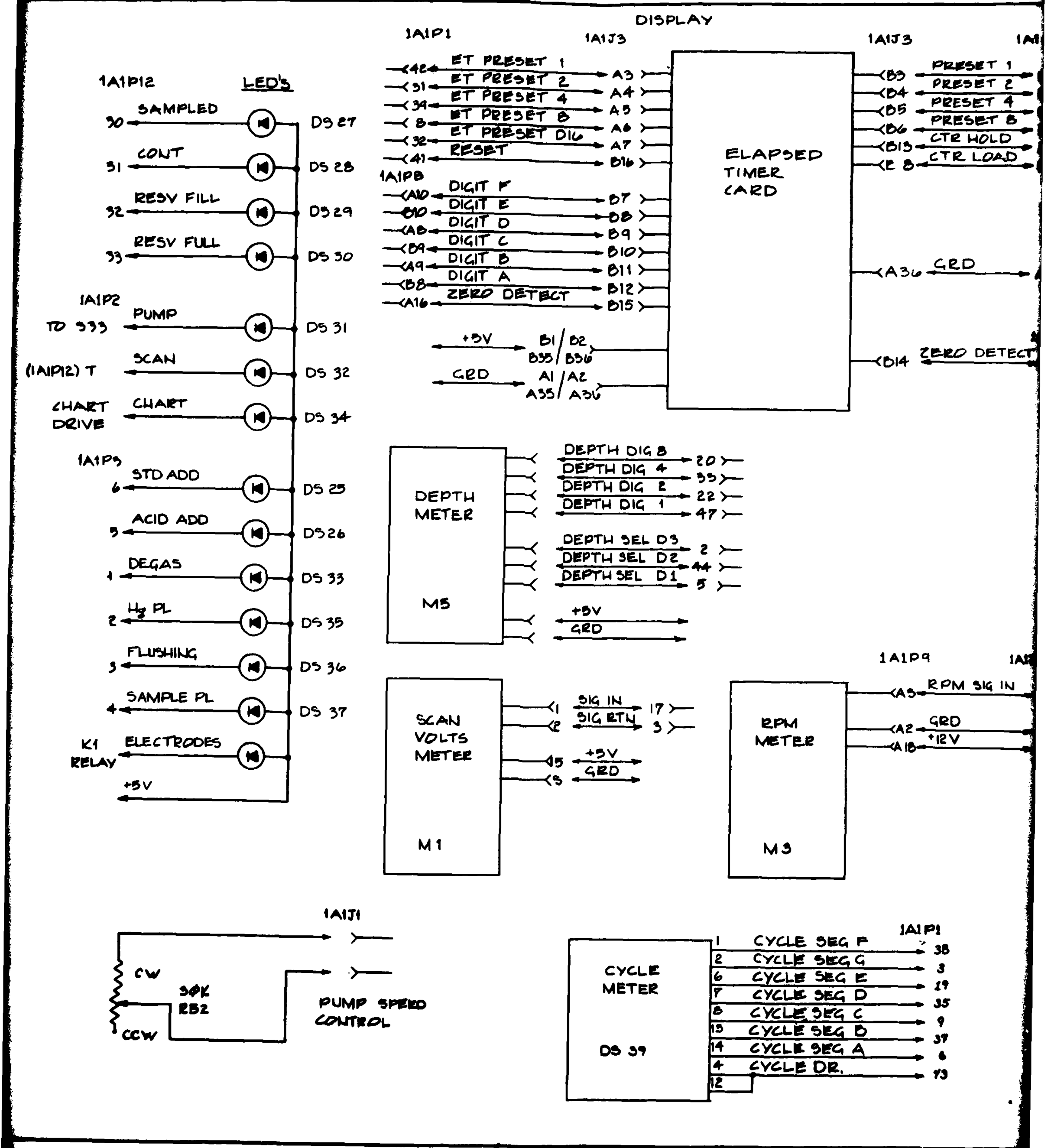
5

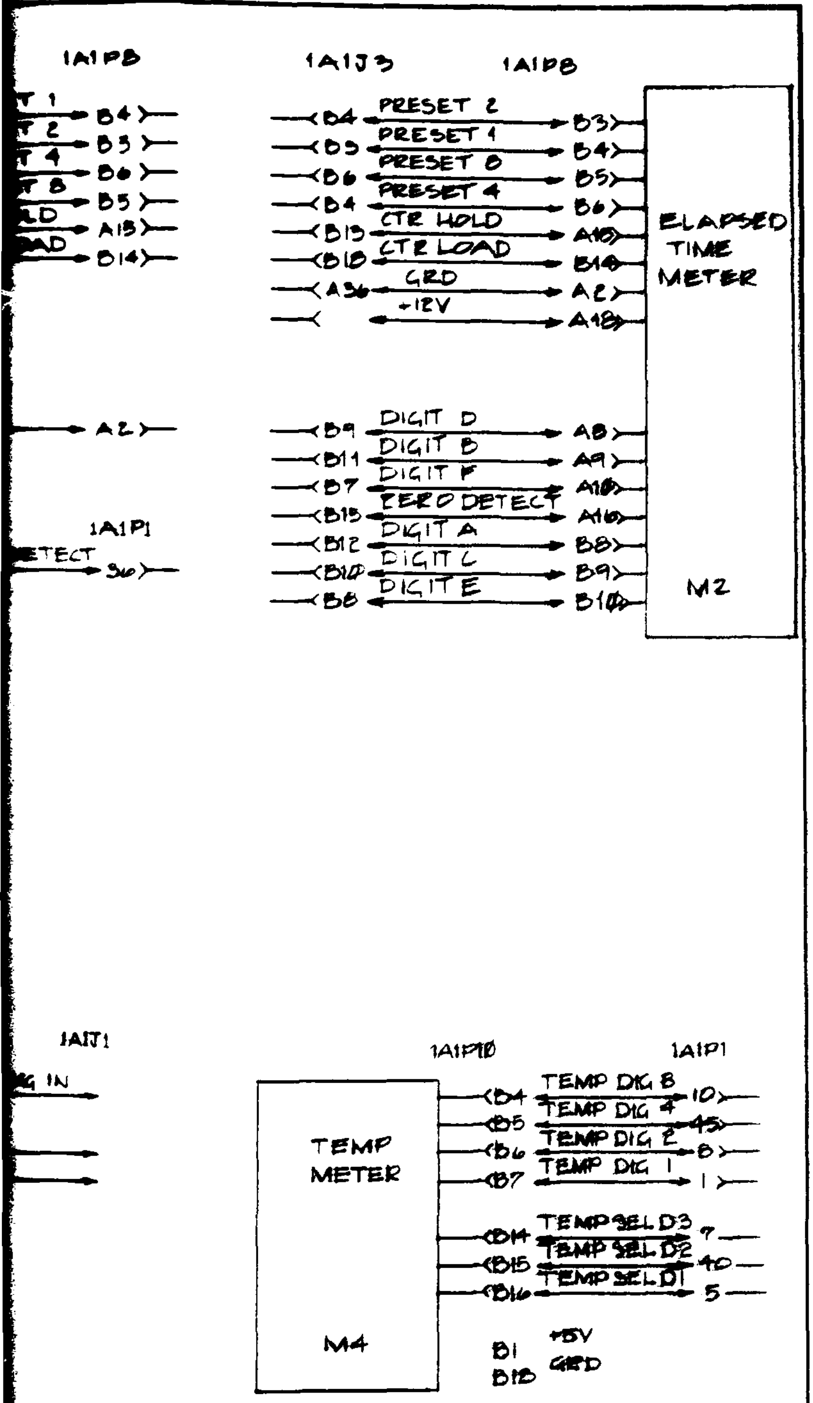


| DESIGN<br>C.CLAVELL | NOSC | DRAFT<br>EMK |
|---|---|---|
| DRW# CC-15 | 9-7-79 | REV# 1 |

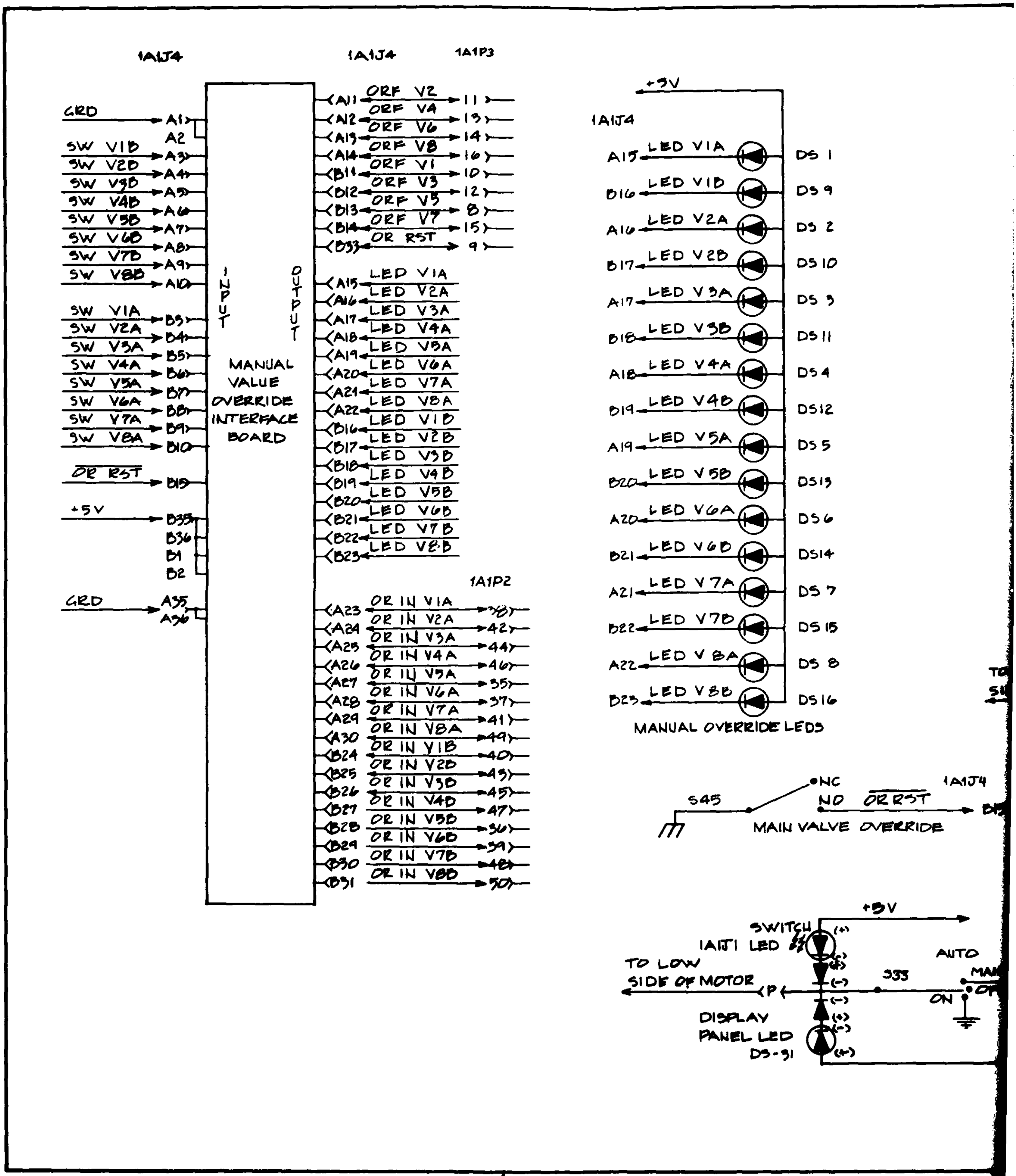
PROGRAM CONTROL &
SELECT SWITCHES
FRONT PANEL I/O

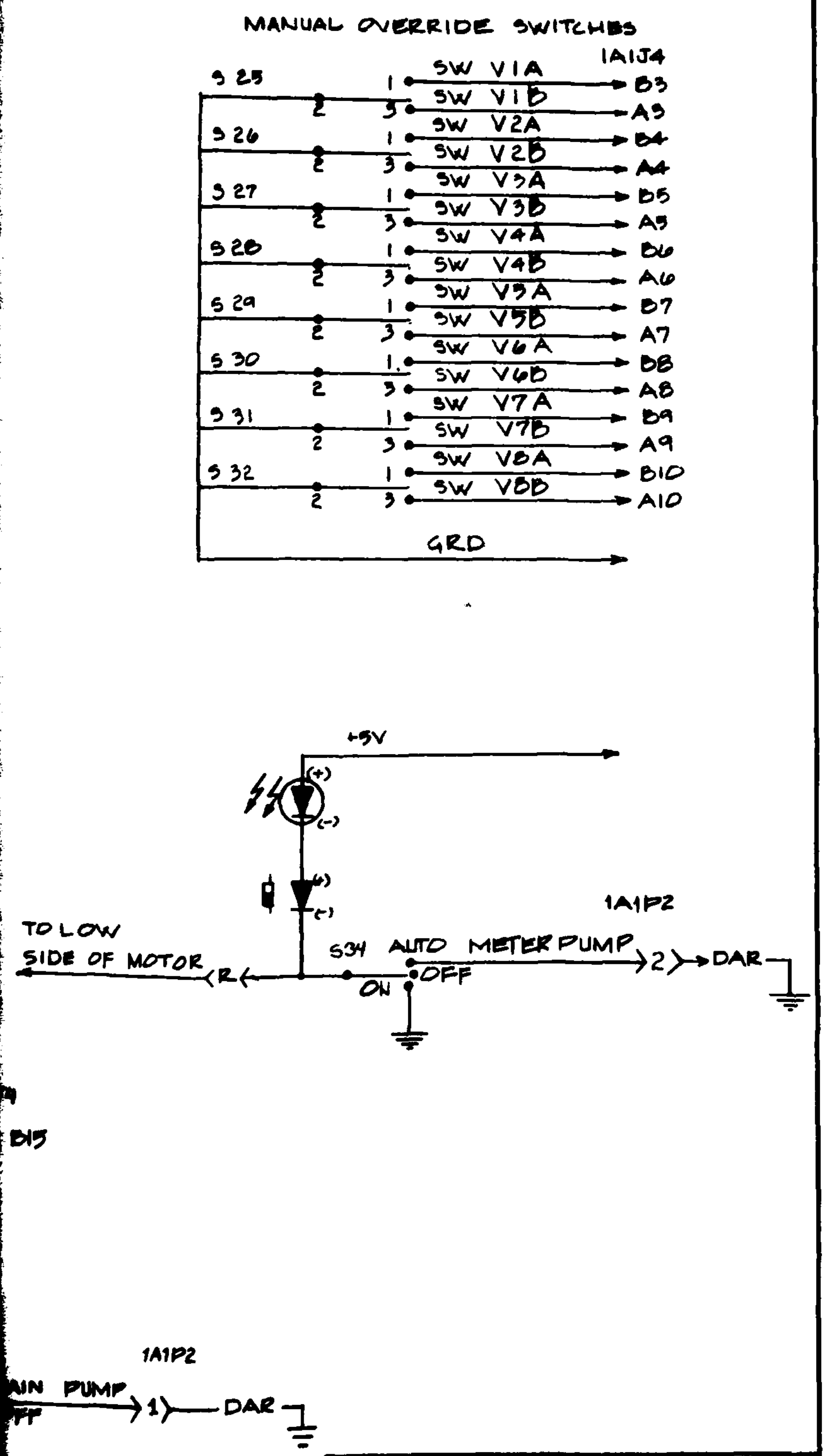
SHT 1 OF 1


DISPLAY
1A1P1
1A1J3
1A1J3
1A1
1A1P12
LED'S
SAMPLED
30
DS 27
CONT
31
DS 28
RESV FILL
32
DS 29
RESV FULL
33
DS 30
1A1P2
PUMP
TO 333
DS 31
SCAN
(1A1P12) T
DS 32
CHART DRIVE
CHART
DS 34
1A1P3
STD ADD
6
DS 25
ACID ADD
5
DS 26
DEGAS
1
DS 33
Hg PL
2
DS 35
FLUSHING
3
DS 36
SAMPLE PL
4
DS 37
K1 RELAY
ELECTRODES
+5V
42
ET PRESET 1
A3
31
ET PRESET 2
A4
34
ET PRESET 4
A5
8
ET PRESET 8
A6
32
ET PRESET DIG
A7
41
RESET
B16
1A1P8
A10
DIGIT F
B7
B10
DIGIT E
B8
A8
DIGIT D
B9
B9
DIGIT C
B10
A9
DIGIT B
B11
B8
DIGIT A
B12
A16
ZERO DETECT
B15
+5V
B1 / B2
B35 / B36
GRD
A1 / A2
A35 / A36
ELAPSED TIMER CARD
B3
PRESET 1
B4
PRESET 2
B5
PRESET 4
B6
PRESET 8
B13
CTR HOLD
E 8
CTR LOAD
A36
GRD
B14
ZERO DETECT
DEPTH METER
M5
DEPTH DIG 8
20
DEPTH DIG 4
33
DEPTH DIG 2
22
DEPTH DIG 1
47
DEPTH SEL D3
2
DEPTH SEL D2
44
DEPTH SEL D1
5
+5V
GRD
SCAN VOLTS METER
M1
1
SIG IN
17
2
SIG RTN
3
15
+5V
5
GRD
1A1P9
1A1
RPM METER
M3
A5
RPM SIG IN
A2
GRD
A18
+12V
1A1J1
CW
50K
R52
CCW
PUMP SPEED CONTROL
CYCLE METER
DS 39
1A1P1
1
CYCLE SEG F
38
2
CYCLE SEG G
3
6
CYCLE SEG E
19
7
CYCLE SEG D
35
8
CYCLE SEG C
9
13
CYCLE SEG B
37
14
CYCLE SEG A
6
4
CYCLE DR.
43
12



| DESIGN<br>L. CLAWELL | NOSC | DRAFT<br>EMK. |
|---|---|---|
| DRW# CC-16 | 9-7-77 | REV# 1 |
| FRONT PANEL I/O<br>DISPLAY | | SHT 1 OF 1 |


1A1J4
1A1J4
1A1P3
GRD A1 A2
SW V1B A3
SW V2B A4
SW V3B A5
SW V4B A6
SW V5B A7
SW V6B A8
SW V7B A9
SW V8B A10
SW V1A B3
SW V2A B4
SW V3A B5
SW V4A B6
SW V5A B7
SW V6A B8
SW V7A B9
SW V8A B10
OR RST B15
+5V B35 B36 B1 B2
GRD A35 A36
INPUT
OUTPUT
MANUAL VALVE OVERRIDE INTERFACE BOARD
A11 ORF V2 11
A12 ORF V4 13
A13 ORF V6 14
A14 ORF V8 16
B11 ORF V1 10
B12 ORF V3 12
B13 ORF V5 8
B14 ORF V7 15
B33 OR RST 9
A15 LED V1A
A16 LED V2A
A17 LED V3A
A18 LED V4A
A19 LED V5A
A20 LED V6A
A21 LED V7A
A22 LED V8A
B16 LED V1B
B17 LED V2B
B18 LED V3B
B19 LED V4B
B20 LED V5B
B21 LED V6B
B22 LED V7B
B23 LED V8B
1A1P2
A23 OR IN V1A 38
A24 OR IN V2A 42
A25 OR IN V3A 44
A26 OR IN V4A 46
A27 OR IN V5A 35
A28 OR IN V6A 37
A29 OR IN V7A 41
A30 OR IN V8A 49
B24 OR IN V1B 40
B25 OR IN V2B 43
B26 OR IN V3B 45
B27 OR IN V4B 47
B28 OR IN V5B 36
B29 OR IN V6B 39
B30 OR IN V7B 48
B31 OR IN V8B 50
+5V
1A1J4
A15 LED V1A DS 1
B16 LED V1B DS 9
A16 LED V2A DS 2
B17 LED V2B DS 10
A17 LED V3A DS 3
B18 LED V3B DS 11
A18 LED V4A DS 4
B19 LED V4B DS 12
A19 LED V5A DS 5
B20 LED V5B DS 13
A20 LED V6A DS 6
B21 LED V6B DS 14
A21 LED V7A DS 7
B22 LED V7B DS 15
A22 LED V8A DS 8
B23 LED V8B DS 16
MANUAL OVERRIDE LEDS
TO
SI
S45
NC
NO
OR RST
1A1J4
B1
MAIN VALVE OVERRIDE
+5V
SWITCH
1A1J1 LED
(+)
(-)
(+)
(-)
TO LOW SIDE OF MOTOR
P
(-)
(+)
(-)
(+)
DISPLAY PANEL LED
DS-31
S33
AUTO
MA
ON
OF



| DESIGN<br>C.CLARK | NOSC | DRAFT<br>EHK. |
|---|---|---|
| DRW# CC-17 | 9-7-79. | REV# 1 |

MANUAL OVERRIDE FRONT PANEL I/O

SHT 1 OF 1

2

ILMED

-8